

ZÁRÓDOLGOZAT

Joó Edvárd
Kotán Péter
Labanc Dániel
2022

Beosztáskészítő webalkalmazás

Joó Edvárd

Kotán Péter

Labanc Dániel

**SZÁMALK-Szalézi Technikum és
Szakgimnázium**

Szoftverfejlesztő

Nyilatkozat

a záródolgozat eredetiségéről

Alulírott Joó Edvárd {anyja neve: Schmidt Ildikó, szem. ig. szám: 741377CE} büntetőjogi és fegyelmi felelősségem tudatában kijelentem és aláírásommal igazolom, hogy a záródolgozat saját munkám eredménye. A felhasznált irodalmi és egyéb információs forrásokat az előírásoknak megfelelően kezeltem, a záródolgozat készítésre vonatkozó szabályokat betartottam.

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

Kijelentem, hogy a záródolgozatom elektronikus változata teljes egészében megegyezik a nyomtatott formával.

Hozzájárulok ahhoz, hogy az érvényben lévő jogszabályok és a Számalk-Szalézi Szakgimnázium belső szabályzata alapján az iskola saját könyvtárában megtekinthető (olvasható) legyen a záródolgozatom.

A záródolgozat titkos/nem titkos.

Budapest, 2022. május 1.

Tanuló aláírása

Nyilatkozat

a záródolgozat eredetiségéről

Alulírott Kotán Péter Balázs {anyja neve: Baló Krisztina, szem. ig. szám: 924182SA} büntetőjogi és fegyelmi felelősségem tudatában kijelentem és aláírásommal igazolom, hogy a záródolgozat saját munkám eredménye. A felhasznált irodalmi és egyéb információs forrásokat az előírásoknak megfelelően kezeltem, a záródolgozat készítésre vonatkozó szabályokat betartottam.

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

Kijelentem, hogy a záródolgozatom elektronikus változata teljes egészében megegyezik a nyomtatott formával.

Hozzájárulok ahhoz, hogy az érvényben lévő jogszabályok és a Számalk-Szalézi Szakgimnázium belső szabályzata alapján az iskola saját könyvtárában megtekinthető (olvasható) legyen a záródolgozatom.

A záródolgozat titkos/nem titkos.

Budapest, 2022. május 1.

Tanuló aláírása

Nyilatkozat

a záródolgozat eredetiségéről

Alulírott Labanc Dániel {anyja neve: Szente Valéria, szem. ig. szám: 492940HE} büntetőjogi és fegyelmi felelősségem tudatában kijelentem és aláírásommal igazolom, hogy a záródolgozat saját munkám eredménye. A felhasznált irodalmi és egyéb információs forrásokat az előírásoknak megfelelően kezeltem, a záródolgozat készítésre vonatkozó szabályokat betartottam.

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

Kijelentem, hogy a záródolgozatom elektronikus változata teljes egészében megegyezik a nyomtatott formával.

Hozzájárulok ahhoz, hogy az érvényben lévő jogszabályok és a Számalk-Szalézi Szakgimnázium belső szabályzata alapján az iskola saját könyvtárában megtekinthető (olvasható) legyen a záródolgozatom.

A záródolgozat titkos/nem titkos.

Budapest, 2022. május 1.

Tanuló aláírása

Tartalom

1	Bevezetés	10
1.1	Elérési út	10
1.2	Téma ismertetése	10
1.3	Előtörténet	10
2	Tervezés, drótvázak	12
3	Felhasználói dokumentáció	15
3.1	A program célja és lényegesebb funkciói	15
3.2	Futtatási környezet	15
3.3	A program fő funkcióinak leírása	16
3.3.1	Bejelentkezés	16
3.3.2	Jelszó visszaállítás.....	16
3.3.3	Szerepkör választás	17
3.3.4	Lenyíló menü / Menüsáv.....	17
3.3.5	Manager menü.....	18
3.3.6	User menü.....	22
4	Fejlesztői dokumentáció	25
4.1	Fejlesztői környezet	25
4.2	Program struktúra	25
4.3	Adatbázis	27
4.3.1	Leírás	27
4.3.2	Funkciók	28
4.3.3	Szerkezet	28
4.3.4	Egyed-kapcsolat diagram	29
4.3.5	Kapcsolati ábra	31
4.3.6	Táblák szerkezete a tábla szintű megszorításokkal	31
4.4	Backend	34
4.4.1	Modellek és Controllerek.....	34
4.4.2	API végpontok	35
4.4.3	Autentikáció/Hitelesítés.....	39

4.5	Frontend	39
4.5.1	Javascript és jQuery.....	39
4.5.2	ajax.js.....	41
4.5.3	Google Charts.....	43
4.5.4	Font Awesome	43
4.5.5	CSS	44
4.5.6	user.js	44
4.5.7	manager.js.....	46
4.5.8	Osztalyok.js	48
4.5.9	Oldalesemenyek.js	49
4.6	Reszponzivitás	50
4.6.1	Mit jelent a reszponzív weboldal?	50
4.6.2	Hogyan vágunk bele a tervezésbe?.....	50
4.6.3	Reszponzív kialakítás.....	51
5	Tesztelés	55
6	Fejlesztési lehetőségek	64
6.1.1	Aktuális ügymenet mentése.....	64
6.1.2	Faliújságon kívül, személyes üzenetek.....	64
6.1.3	A felhasználói felületet React-tel elkészíteni	64
6.1.4	Éjszakai / Nappali mód	64
7	Irodalomjegyzék	1

Ábrajegyzék

1. ábra - Admin oldal	12
2. ábra - Bejelentkezés	12
3. ábra - Manager oldal	13
4. ábra - Szerepkör választó	13
5. ábra - User oldal.....	14
6. ábra - Bejelentkezés	16
7. ábra - Elfelejtett Jelszó.....	16
8. ábra - Jelszó Visszaállítás	16
9. ábra - Szerepkör választó	17
10. ábra - Manager Profil lenyíló menü.....	17
11. ábra - User Profil menüsáv	17
12. ábra - Profil Főmenü	18
13. ábra - Manager Főmenü	18
14. ábra - Gyors módosítás	18
15. ábra - Munkakörök kezelése	19
16. ábra - Műszaktípus naphoz rendelése	19
17. ábra - Napi munkaerőigény meghatározása.....	20
18. ábra - Alkalmazott beosztása	20
19. ábra - Beosztás megtekintése	21
20. ábra - Alkalmazottak munkakör szerinti eloszlása - Diagram	21
21. ábra - Manager faliújság	22
22. ábra - Heti óraszám - Diagram.....	22
23. ábra - Profiladatok	23
24. ábra - Nem érek rá	24
25. ábra - Beosztás	24
26. ábra - Egyed-kapcsolat diagram	30
27. ábra - Kapcsolati ábra	31
28. ábra - User főtároló grid elrendezés 1300px felett	51
29. ábra - User oldal grid elrendezés 1300px felett	52
30. ábra - User főtároló grid elrendezés 1300px alatt.....	53
31. ábra - User oldal grid elrendezés 1300px alatt	53
32. ábra - Mobilnézet menüsáv	54

33. ábra - AlkalmazottController Teszt	61
34. ábra - BejelentkezesiAdatokController Teszt	61
35. ábra - BeosztasController Teszt	61
36. ábra - FaliujsagController Teszt	61
37. ábra - MunkakorController Teszt	62
38. ábra - MuszakEloszlasController Teszt	62
39. ábra - MuszakTipusController Teszt	62
40. ábra - NapokController Teszt	62
41. ábra - NemDolgoznaController Teszt.....	63
42. ábra - SzabadsagController Teszt	63

1 Bevezetés

1.1 Elérési út

A weboldal [ide kattintva](#) érhető el.

A webhely URL címe: <http://82.150.42.74:8087/>

A felhasználóhoz egységesen „asd123” a bejelentkezési jelszó. A webalkalmazásban három különböző szerepkörrel rendelkező felhasználót különböztetünk meg, melyek a következő belépési adatokkal rendelkeznek:

Szerepkör	Felhasználónév
Adminisztrátor	40034
Manager	40015
User	40010

1.2 Téma ismertetése

Szakdolgozatunk fő célja egy olyan beosztáskezelő webalkalmazás, mely egyetlen felületre sűríti a szervezett munkavégzéshez szükséges kommunikációs csatornákat. Igyekeztünk olyan adaptív webalkalmazást készíteni, mely a legtöbb munkahelyen alkalmazható. Webes alkalmazásunkat egy fiktív kávézóhoz, a *Chill out Café*-hoz készítettük, melyben külön felületet kap az üzletvezető és az alkalmazottak is, továbbá az adminisztrátoroknak olyan oldalt készítettünk, amely minden egyéb adatbázis-kezelő nyelv ismerete nélkül használható.

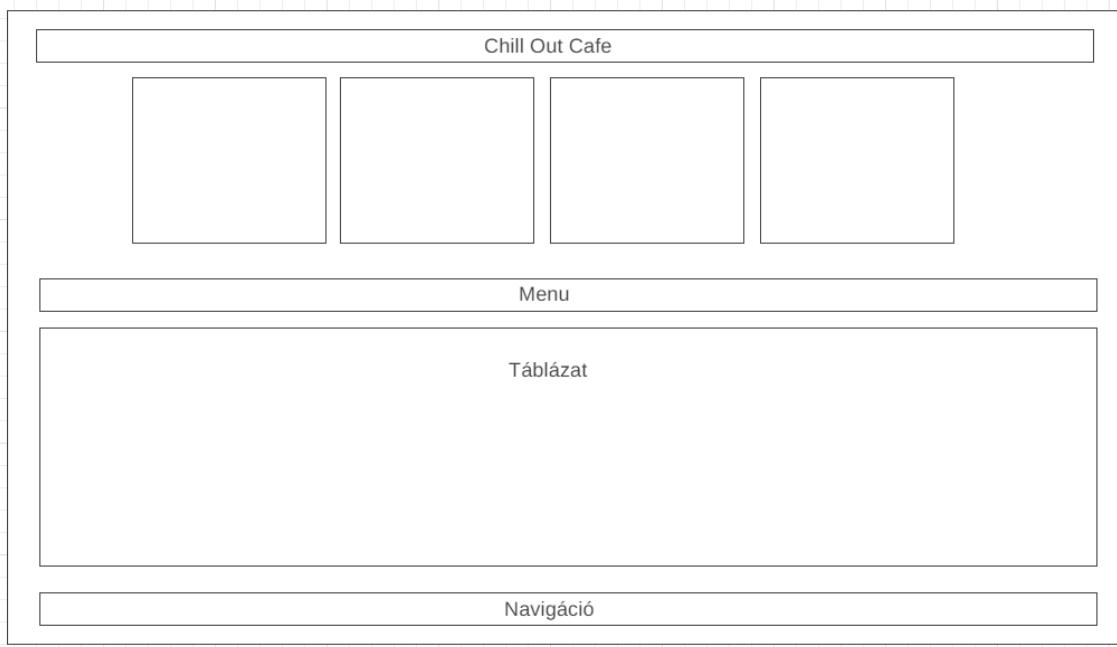
1.3 Előtörténet

Mivel már mindhárman megfordultunk a munkaszférában, így több beosztási formával is találkoztunk, nem mindegyik nyerte el tetszésünket. Bizonyos munkahelyek egyszerűen a közösségi oldalakat használják, a munkáltató pedig egy Excel táblázatban készíti el fáradalmas úton a munkavállalóinak napirendjét. Máshol találkoztunk már kézzel írt beosztásokkal, ahol a munkáltató egyesével küldte el a dolgozónak a rájuk tartozó műszakokat.

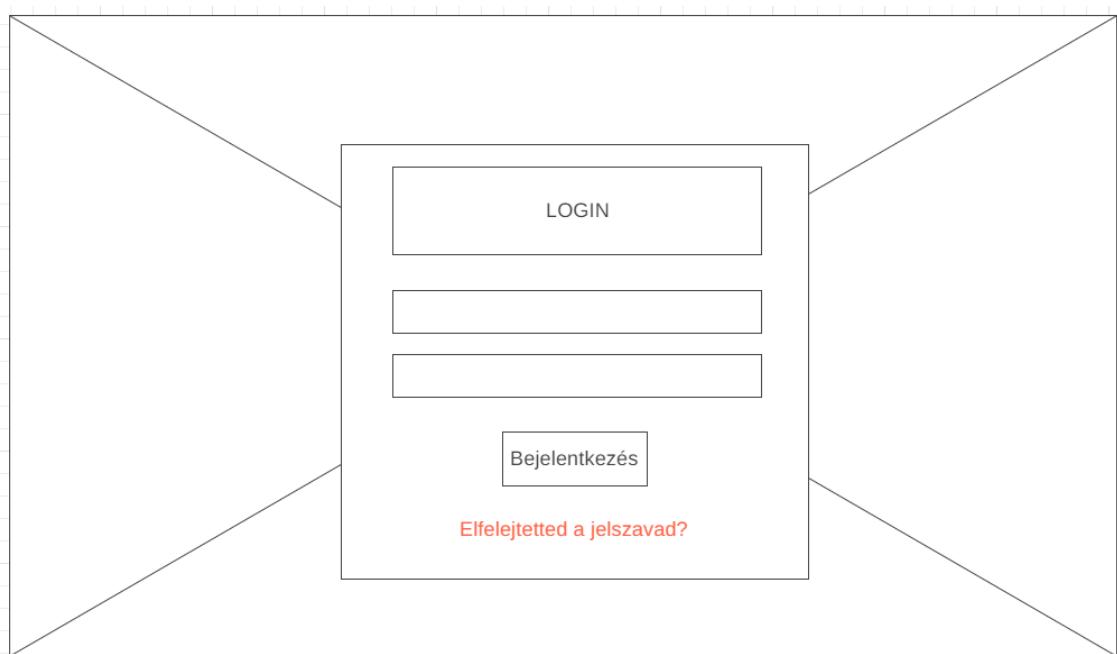
Fő motivációink a téma választásban tehát az volt, hogy a már tapasztalt felületeket, megoldásokat túlszárnyaljuk, és minden olyan hiányosságot, amiket korábbi

munkahelyeinken találkoztunk, betöltsünk. Ezzel kedvezve a munkáltatónak is, hogy könnyebben el tudja végezni a beosztás elkészítését, valamint a munkavállalónak is, hogy minden, ami a munkájához kapcsolódik egy helyen lásson.

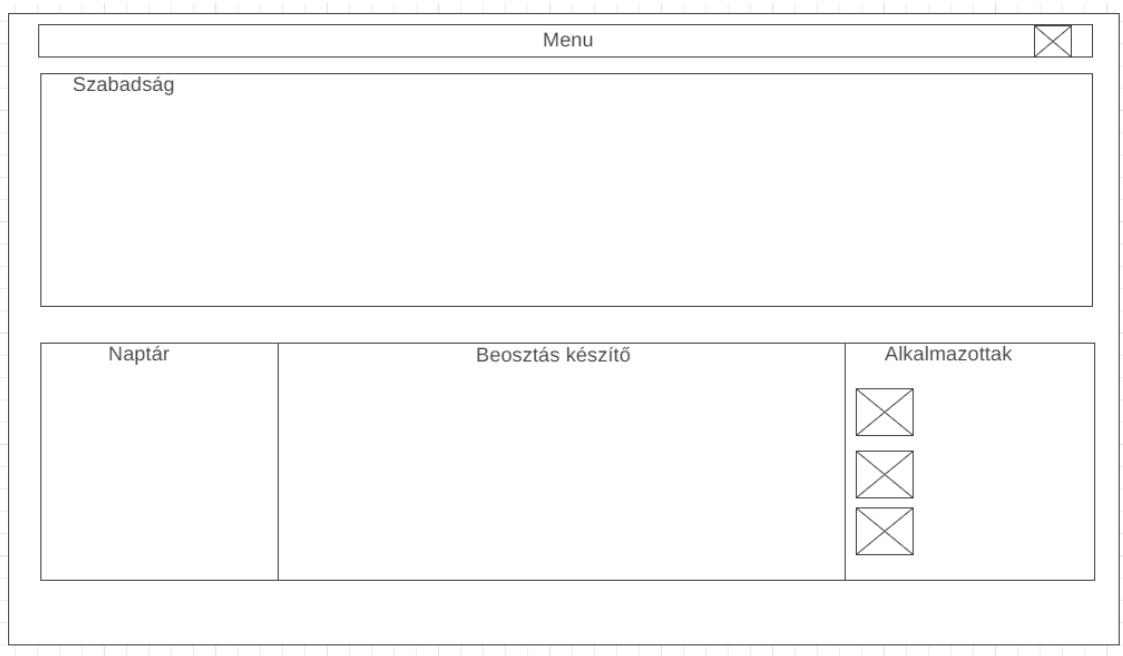
2 Tervezés, drótvázak



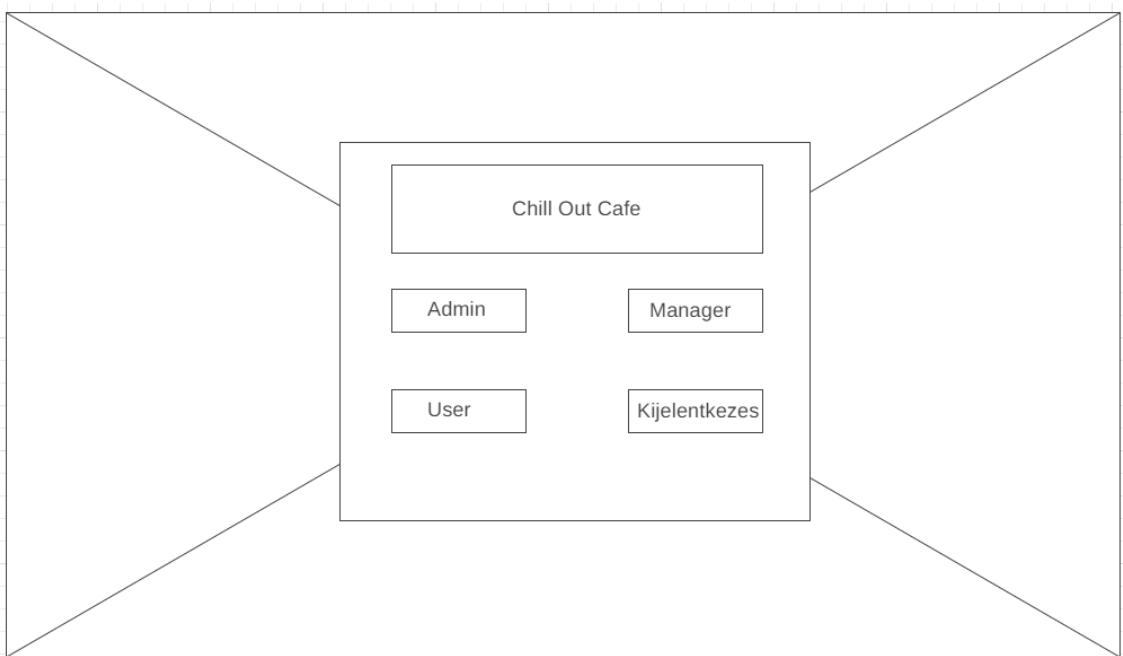
1. ábra - Admin oldal



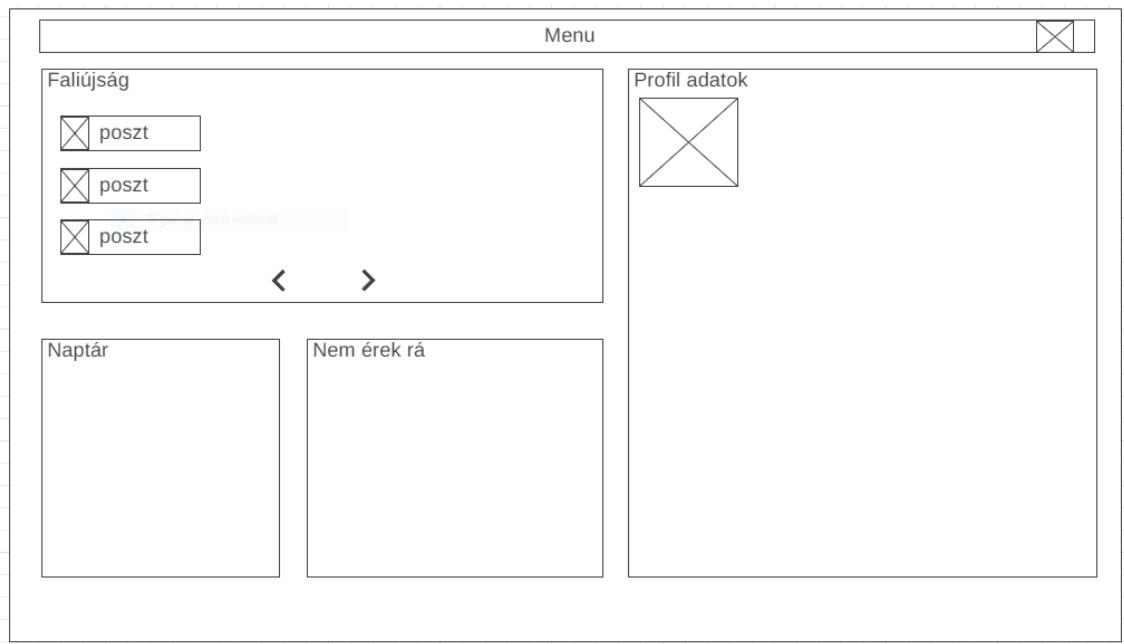
2. ábra - Bejelentkezés



3. ábra - Manager oldal



4. ábra - Szerepkör választó



5. ábra - User oldal

3 Felhasználói dokumentáció

3.1 A program célja és lényegesebb funkciói

A *Chill out Café* oldala lehetővé teszi a munkáltató számára, hogy megtekintse és kezelje az alkalmazottainak adatait, új munkakört vegyen fel, a meglévőket módosítsa, illetve, hogy elkészítse és módosítsa a heti munkabeosztásokat, illetve a munkára vonatkozó statisztikákat megtekintse. A munkavállalóknak lehetőségük nyílik saját adataik, valamint a beosztásuk megtekintésére. Ezen felül pedig minden félnek létrehoztunk egy faliújságot, ahol a munkavégzéshez szükséges információkat tudják megosztani egymással. Továbbá az adminisztrátornak olyan felületet készítettünk, amely lehetőséget kínál a program mögött lévő adatbázisba új adatok felvételére, a meglévők módosítására és törlésére.

3.2 Futtatási környezet

Amint felmerült az ötlet, hogy beosztás tervező/készítő applikációt szeretnénk készíteni, egyből világossá vált számunkra, hogy egy webalkalmazás lenne a legcélravezetőbb. Egyrészről fontos volt, hogy a munkahely alkalmazottai telefonon, bármikor meg tudják nézni a beosztásukat anélkül, hogy telepíteniük kéne bármilyen alkalmazást, más részről pedig szerettük volna, hogy a menedzserek telefonról, tabletről is képesek legyenek a munkájukat végezni. Igaz, az alkalmazás használata ügyviteli szempontból átláthatóbb és olvashatóbb nagyobb felbontású monitoron (ez kifejezetten igaz az adminisztrátori munkára), de minden feladatot képesek vagyunk elvégezni kisebb kijelzőkön is. Ezen kívül gondoltunk arra is, hogy platform független legyen és bármilyen eszkösről használni tudjuk az oldalakat, így nem kellett minden platformra külön applikációt készíteni. Azt gondolom, manapság egy felhasználó szívesebben látogat meg egy oldalt, mintsem tölti le az alkalmazás szoftverét, ha az egy belsős rendszer részét képezi.

A weboldal megnyitására Mozilla Firefox, vagy Google Chrome webböngészők legfrissebb verziói a legalkalmasabbak. Az oldalak nem követelnek meg az átlagosnál nagyobb sávszélességű internetkapcsolatot.

3.3 A program fő funkcióinak leírása

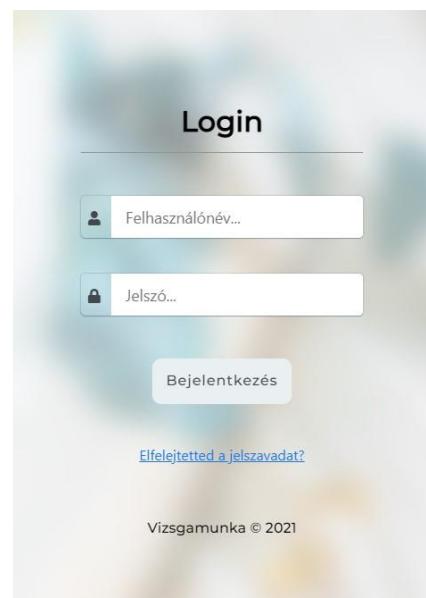
3.3.1 Bejelentkezés

Az oldal funkcióit kizárolag a bejelentkezett felhasználók érhetik el. minden egyes alkalmazott beiktatáskor kap egy azonosítót, (továbbiakban ez fogja képezni a felhasználónévét) és egy hozzá tartozó kezdeti jelszót. Ezzel a két adattal tud mindenki bejelentkezni a saját személyes felületére. A kezdeti jelszót első belépést követően meg kell változtatni. Mivel a belépési adatokat az adminisztrátor tölti fel az adatbázisba, így az oldal regisztrációs lehetőséggel nem rendelkezik.

Bejelentkezést követően a *User* szerepkörrel rendelkező felhasználókat a rendszer automatikusan a saját felületükre irányítja át, a Manager vagy Adminisztrátor szerepkörrrel rendelkező felhasználók pedig a szerepkör választó menübe kerülnek.

3.3.2 Jelszó visszaállítás

Ha a felhasználó elfelejtette volna a jelszavát, van lehetősége az email címére kérni egy jelszó visszaállító linket, amelyre kattintva be tud állítani egy teljesen új jelszót.



6. ábra - Bejelentkezés

Chill Out Café

Elfelejtette a jelszavát?

Adja meg a regisztrációkor használt e-mail címét és elküldjük a jelszó visszaállításához szükséges utasításokat.

Biztonsági okokból NEM tároljuk jelszavát. Biztos lehet benne, hogy soha nem küldjük el jelszavát e-mailben!

Email:

adminandras@chilloutcafe.hu

Küld

7. ábra - Elfelejtett Jelszó

Chill Out Café

Jelszó visszaállítás

Email:

adminandras@chilloutcafe.hu

Jelszó:

.....

Jelszó újra:

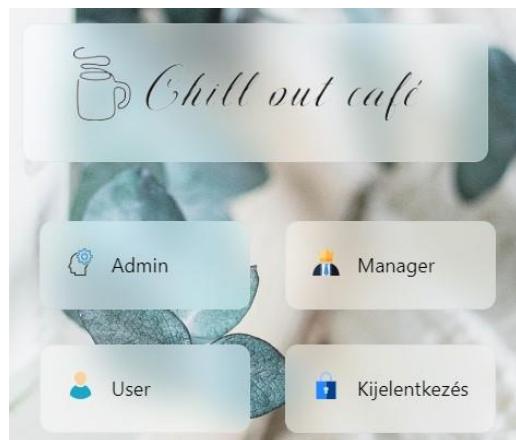
.....

Küld

8. ábra - Jelszó Visszaállítás

3.3.3 Szerepkör választás

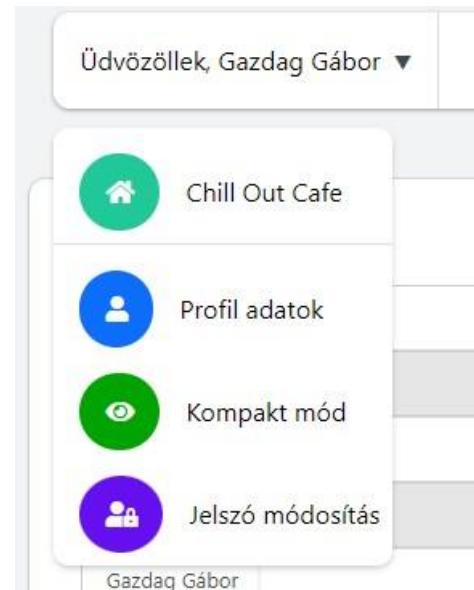
A szerepkör választó oldal kizárolag a Manager és Adminisztrátor szerepkörrel rendelkező felhasználóknak érhető el. Mindkét felhasználó ki tudja választani saját szerepkörén felül a személyes adatait tartalmazó User szerepkört is. A gombokra kattintva a felhasználó átkerül a kiválasztott szerepkör oldalára. Ezen felül itt jelenik meg a Kijelentkezés menüpont, ami kijelentkezteti a felhasználót és visszairányítja a bejelentkező felülethez.



9. ábra - Szerepkör választó

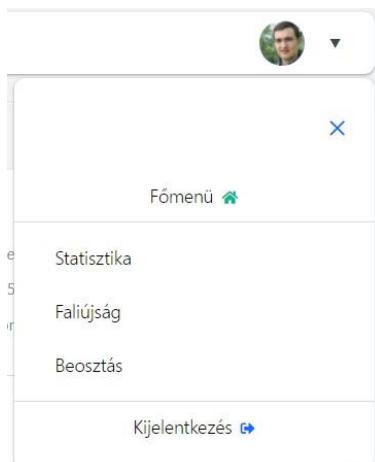
3.3.4 Lenyíló menü / Menüsáv

A weboldalon kétféle menüt különböztetünk meg. A Profilmenüben tudja elérni a felhasználó a személyes adatait, itt találja a jelszó módosítást is, illetve be tudja kapcsolni a weboldalon a kompakt módot. Ezt a menüt a Manager menüben bal oldalra, egy lenyíló ablakra helyeztük, míg a User menüben középre, menüsávként pozicionáltuk. A lenyíló menü ezen felül még tartalmaz egy főoldal gombot, amely visszanavigálja a felhasználót a főoldalra.



10. ábra - Manager Profil lenyíló menü

A Főmenü minden oldal esetében a felület jobb oldalán jelenik meg. Ezek a menük tartalmaznak lenyitható almenüket is. Itt is megtalálható a Főoldal gomb, illetve ide került be a Kijelentkezés, mely a felhasználót a bejelentkezási felületre navigálja vissza.



12. ábra - Profil Főmenü

13. ábra - Manager Főmenü

3.3.5 Manager menü

- Alkalmazottak adatainak kezelése, módosítása

Az *Alkalmazottak* menüpontban a menedzser megtekintheti táblázatos formában az összes beosztottját. Egy alkalmazottra kattintva a felhasználó kiválaszthatja, hogy gyors módosítást szeretne végezni, ekkor csak a táblázatban megjelent alap adatok módosítására (név, beosztás, lakcím, elérhetőség, e-mail) van lehetőség, vagy pedig az alkalmazott teljes profilját szeretné megtekinteni. Utóbbi lehetőséget ad az összes adat módosítására.

Név	Beosztás	Lakcím	Elérhetőség	E-mail
Kis Pista	Felszolgáló	1089 Budapest, Német u. 3	06701111111	kispista@chilloutcafe.hu
Nagy Mari	Felszolgáló	1011 Budapest, Alfa u. 1	06702222222	nagymari@chilloutcafe.hu
Hosszu Iván	Felszolgáló	1045 Budapest, Elit u. 12	06703333333	hosszuivan@chilloutcafe.hu
Rövid Peti	Felszolgáló	1123 Budapest, Fiaskó u. 144	06704444444	rovidpeti@chilloutcafe.hu
Rövid Peti	Adminisztrátor	1123 Budapest, Fiaskó u. 144	06704444444	rovidpeti@chilloutcafe.hu
Aranyos Aranka	Felszolgáló	1211 Budapest, Lakóház u. 6	06705555555	aranyosaranka@chilloutcafe.hu
Eszes Dani	Pultos	1055 Budapest, Infravörös u. 4	06706666666	eszsedani@chilloutcafe.hu
Izsákos Imre	Pultos	1012 Budapest Etele út 3	06704891264	izsakosimre@chilloutcafe.hu
Kenyér Ádám	Pultos	1023 Budapest Felső u. 9°	06704112561	kenyeradam@chilloutcafe.hu
Csőrő Lili	Pultos	1034 Budapest Izé u. 12	06704123523	csorolali@chilloutcafe.hu

14. ábra - Gyors módosítás

- Munkakörök kezelése, új munkakör felvétele

Ebben a menüpontban a menedzsernek lehetősége nyílik a regisztrált munkakörök, illetve az adott munkakörben dolgozó alkalmazottak megtekintésére, valamint új munkakör felvételére. Mivel ez a felület nem az alkalmazottak adatainak kezelésére szolgál, így az alkalmazottra vonatkozóan csak három alapadatot jelenítünk meg (név, telefonszám, e-mail cím).

15. ábra - Munkakörök kezelése

- Beosztás elkészítése, módosítása, megtekintése

A beosztás elkészítését a menedzser három lépésben tudja elvégezni. Ehhez az főoldalra beillesztettünk egy belső hivatkozásokkal ellátott útmutatót, melyen egyszerűen, lépésről lépésre leírjuk, mik a beosztáskészítés lépései.

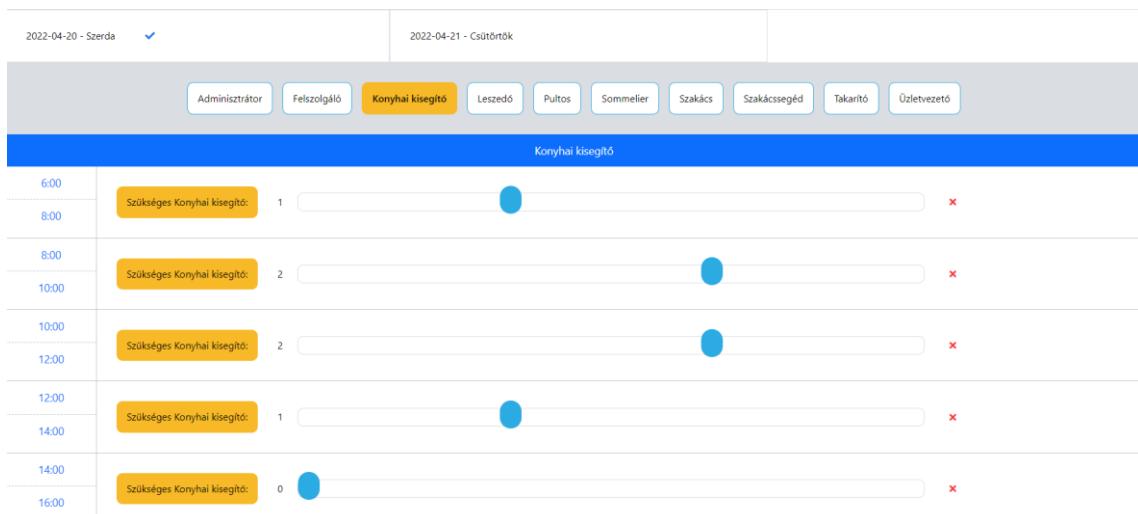
a) Műszaktípus naphoz rendelése

Elsősorban ki kell választanunk a főoldalon elhelyezett naptáróból, hogy melyik nap műszakját akarjuk elkészíteni. A már előre felvett munkarendekből (A, B, C) választhatunk. Ezek a munkarendek más nyitvatartást jelölnek, illetve más a műszakintervallumok leosztása is. Azt a napot, amihez már rendeltünk munkarendet a naptárban zölddel jelezük. Munkarend felvétele után lehetőségünk van az adott naphoz tartozó hozzárendelés törlésére, ezek után pedig másik munkarend hozzáadására is.

b) Napi munkaerőigény meghatározása

16. ábra - Műszaktípus naphoz rendelése

A Beosztás menüpont *Napi munkaerőigény* almenüjére kattintva megjelennek azok a napok, amelyekhez már hozzárendeltünk műszaktípushat. Itt műszakonként tudjuk beállítani, hogy az adott munkakörből hány alkalmazottra van szükségünk. Ezt egy egyszerű csúszka (range slider) segítségével tehetjük meg. A munkaerőigény meghatározása után szükséges az adott nap véglegesítésére.



17. ábra - Napi munkaerőigény meghatározása

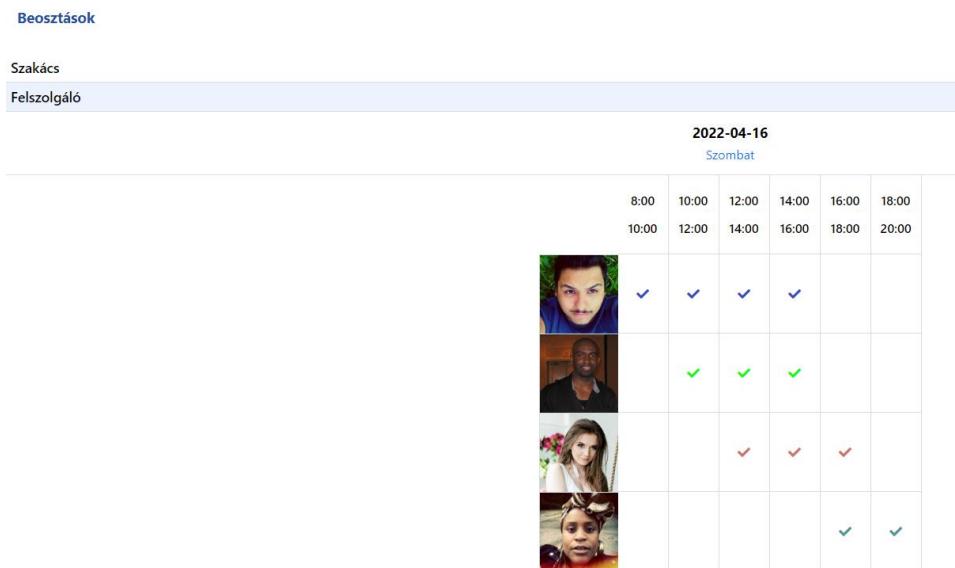
c) Alkalmazottak beosztása

Harmadik lépésként a főoldalon ismét ki kell választanunk a beállítani kívánt napot a naptáróból. Azonban, ha már hozzárendeltünk munkaerőigényt, úgy az oldal listázza azokat a műszakokat, amikhez szükséges beosztottat rendelnünk. Ezeket a műszakokat szürkén jelöli azzal a számmal, ahány munkaerőigényt beállítottunk. Az oldal jobb oldalán műszakra lebontva találjuk az összes alkalmazottat, amelyeket egyszerűen behúzhatunk a kívánt műszakba.

18. ábra - Alkalmazott beosztása

d) Beosztás megtekintése

A Beosztás menüpont Beosztás megtekintése almenüjébe lépve megtekinthetjük az adott hét beosztásait munkakörökre lebontva.



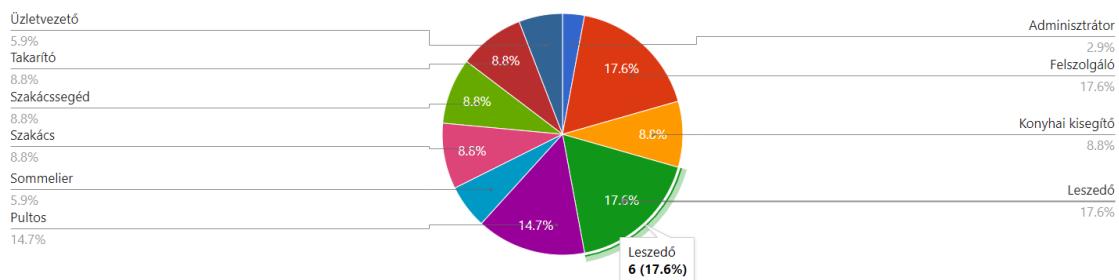
19. ábra - Beosztás megtekintése

Leszedő

- **Statisztikák megtekintése**

Az oldalon a menedzseléshez szükséges adatokról készítettünk statisztikát, és jelenítettük meg a felhasználó számára. A szabadság statisztika a főoldalon található, ahol hónapra lebontva egy idővonalas Google Chart segítségével tekinthetjük meg az alkalmazottak elfogadott szabadságkérelmet.

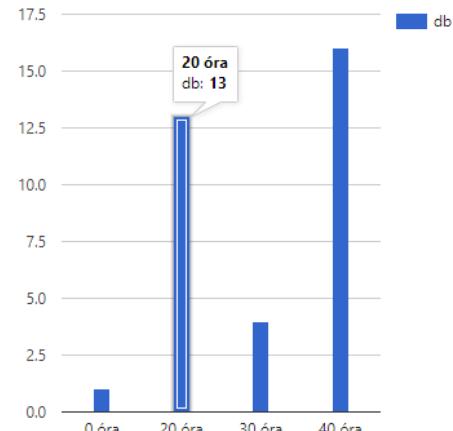
További statisztikákat a főmenü Egyéb menüpont Statisztika almenüjében érhetünk el. Itt kördiagram segítségével ábrázoltuk munkakörökre lebontva az alkalmazottakat százalékosan, illetve létszám szerint.



20. ábra - Alkalmazottak munkakör szerinti eloszlása - Diagram

Megjelenítjük ezen felül az alkalmazásban lévő beosztottak óraszám szerinti eloszlását is, hiszen fontos, hogy a szükséges összes óraszámot arányosan lefedjék az alkalmazottak heti óraszámai.

Továbbá itt is megtalálható a főoldalon megjelenített *Szabadság* statisztika is.



22. ábra - Heti óraszám - Diagram

21. ábra - Manager faliújság

- Faliújság kezelése

Az *Egyéb* menüpont *Faliújság* almenüjében lehetőségünk nyílik új faliújság posztok felvételére, valamint a már publikussá tett posztok módosítására és törlésére. A felvitt poszt minden regisztrált felhasználó számára azonnal láthatóvá válik ezzel is elősegítve a munkához szükséges gyors kommunikációt.

3.3.6 User menü

- Profiladatok megtekintése

A felhasználó itt tudja megtekinteni személyes adatait. Ezen adatok módosításához kizárolag a menedzser és az admin rendelkezik jogosultsággal.

Profilkép

Gazdag Gábor

Üzletvezető

1062 Budapest Van u. 2

06708912645

gazdaggabor@chilloutcafe.hu

Profil adatok

Név:	Gazdag Gábor
Lakcím:	1062 Budapest Van u. 2
Születési Dátum:	1982-10-13
Adóazonosító:	7123456789
TAJ:	89234567
Elérhetőség:	06708912645
E-mail:	gazdaggabor@chilloutcafe.hu
Munkakör:	Üzletvezető
Heti óraszám:	0
Munkaviszony kezdete:	2021-03-15
Munkaviszony Vége:	-

23. ábra - Profiladatok

- Faliújság megtekintése

A User szintű felhasználó ugyanazt a faliújság felületet látja, amit a menedzser annyi különbséggel, hogy a Usernek nem jelenik meg az *új bejegyzés* gomb.

- „Nem érek rá” leadása

Az alkalmazottnak a főoldalon megjelenő naptáron kell kiválasztania azokat a napokat, mikor ráérési igényt szeretne leadni. Amennyiben a választott naphoz még nincs műszaktípus hozzárendelve, úgy egy információs ablak jelenik meg a felhasználónak. Egyéb esetben pedig a listázott műszakokból ki tudja választani azt, amelyik számára nem megfelelő.

The left side shows a 4x7 grid of numbers from 1 to 30. The 16th of April is highlighted with a green rounded rectangle. Above the grid, the date '2022. Április.16.' is displayed. The top bar also shows the date '2022. Április.16.' and the time '21:34:24'.

The right side shows a list of time intervals with checkboxes:

- 8:00 - 10:00
- ~~10:00 - 12:00~~
- ~~12:00 - 14:00~~
- 14:00 - 16:00
- 16:00 - 18:00
- 18:00 - 20:00

At the bottom are two buttons: a green checkmark icon and a red trash bin icon.

24. ábra - Nem érek rá

- Beosztás megtekintése

Minden User szerepkörrel rendelkező alkalmazott a saját beosztását tudja megtekinteni, amit a főmenü *Beosztás* menüpontjában ér el. Mivel előfordulhat, hogy napoként változik a műszaktípus, ezért a beosztás napi lebontásban készül el. Az egyes napokon zöld négyzet jelzi, az adott alkalmazott mikorra lett beosztva.

2022-01-10		2022-01-11	2022-04-16	2022-04-21
6:00	8:00			
8:00	10:00			
10:00	12:00			
12:00	14:00			
14:00	16:00			
16:00	18:00			
18:00	20:00			
20:00	22:00			

25. ábra - Beosztás

4 Fejlesztői dokumentáció

4.1 Fejlesztői környezet

Az alkalmazás szerver-kliens alapon működik, ami az erőforrások jobb kihasználását teszi lehetővé. Ehhez szükség van egy szerver oldali programozási nyelvre, mi esetünkben PHP-ra. A RESTful tervezési feltételeinek megfeleléséhez és a könnyebb kezelhetőség, átláthatóság illetve biztonság érdekében, a Laravel keretrendszert használtunk. A keretrendszer kiszolgálásához szükséges szervernek Apache webszervert vettünk igénybe, valamint MySQL adatbázist. Ezek mind futtathatók Windows és Linux rendszereken is. A kliens oldali programozást Javascript-el valósítottuk meg. A fejlesztés Visual Studio Code-ban történt, verziókövetésre pedig a Githubot alkalmaztuk.

4.2 Program struktúra

Beosztás

- app
 - Http
 - Controllers
 - Auth
 - HitelesitesController.php
 - JelszoVisszaAllitasController.php
 - AlkalmazottController.php
 - BejelentkezesiAdatokController.php
 - BeosztasController.php
 - FaliujsagController.php
 - MunkakorController.php
 - MuszakEloszlasController.php
 - MuszakTipusController.php
 - NapiMunkaeroIgenyController.php
 - NapokController.php
 - NemDolgoznaController.php
 - StatisztikaController.php
 - SzabadsagController.php
 - Middleware
 - Authenticate.php
 - IsAdmin.php
 - IsManager.php
 - RedirectIfAuthenticated.php
 - RedirectIfNotAuthenticated.php
 - Role.php
 - Models

- Alkalmazott.php
 - BejelentkezesiAdatok.php
 - Beosztas.php
 - Faliujsag.php
 - Munkakor.php
 - MuszakEloszlas.php
 - MuszakTipus.php
 - NapiMunkaeroIgeny.php
 - Napok.php
 - NemDolgozna.php
 - Szabadsag.php
- database
 - factories
 - migrations
 - seeders
- public
 - css
 - admin
 - admin.css
 - global
 - global.css
 - login
 - forgetpassword.css
 - login.css
 - resetpassword.css
 - managermenu
 - managermenu.css
 - usermenu
 - usermenu.css
 - js
 - admin
 - admin.js
 - global
 - Ajax.js
 - Oldalesemenyek.js
 - Osztalyok.js
 - managermenu
 - manager.js
 - user
 - user.js
 - pictures
 - node_modules
- resources
 - views
 - admin

- admin.php
- login
 - forgetpassword.php
 - login.php
 - resetpassword.php
- managermenu
 - managermenu.php
- usermenu
 - usermenu.php
- welcome.blade.php
- routes
 - auth.php
 - web.php
- tests
- vendor

4.3 Adatbázis

4.3.1 Leírás

A Chill out Café adatbázis egy étterem munkakörönkénti beosztását készíti el a soron következő hétre.

Az adatbázisban kezeljük az alkalmazottak adatait. minden alkalmazottat munkakör alapján veszünk fel. Az étteremben minden munkakör különböző, de az alkalmazottaknál lehet azonos. A munkakörökhez választunk feletteseket, aki szintén dolgozik. A felettesek a saját munkakörükön belül kerülnek ki. A munkakörök alkalmazottai más óraszámmal vehetők fel.

A műszakok előre meghatározott intervallumok. minden műszakhoz szükséges meghatározni, hogy melyik műszaktípusa. Az étteremben lehetőségünk van arra, hogy az adott hét napjaihoz hozzárendeljük a műszaktípust. Egy napon csak azonos műszaktípusú műszakokra van lehetőség. A munkaerőigény határozza meg a szükséges munkaköröket az adott műszakokhoz. Dátum szerint meghatározzuk, hogy melyik munkakörből, melyik műszakban, mennyi dolgozóra van szükség.

Az alkalmazottak jelzik, hogy mikor nem dolgoznának, ebbe bele tartozhat a szabadság is (fizetett, beteg, nem fizetett). Ennek nem kötelező teljesülnie. Legkésőbb a szabadság dátumától egy hónappal korábban meg kell adni a szabadsági kérelmet. Nyilvántartjuk, hogy melyik dolgozó nem beoszthat, szabadság vagy betegség miatt. A beosztás a munkaerőigény igény alapján jön létre.

4.3.2 Funkciók

- Munkakörök felvétele, módosítása, törlése
- Munkaköri felettesek meghatározása
- Alkalmasztottak felvétele, módosítása, törlése
- Műszaktípus hozzáadása, módosítása, törlése
- Műszaktípus naphoz rendelése, eltávolítása, állapot módosítása
- Műszakeloszlás módosítása, törlése
- Munkaerő igény leadása, módosítása, törlése
- "Nem dolgozna" leadása, törlése
- Beosztás készítése, módosítása, törlése
- Faliújság bejegyzések felvétele, módosítása
- Szabadság kérelem felvétele, elfogadása, törlése
- Statisztika megtekintés

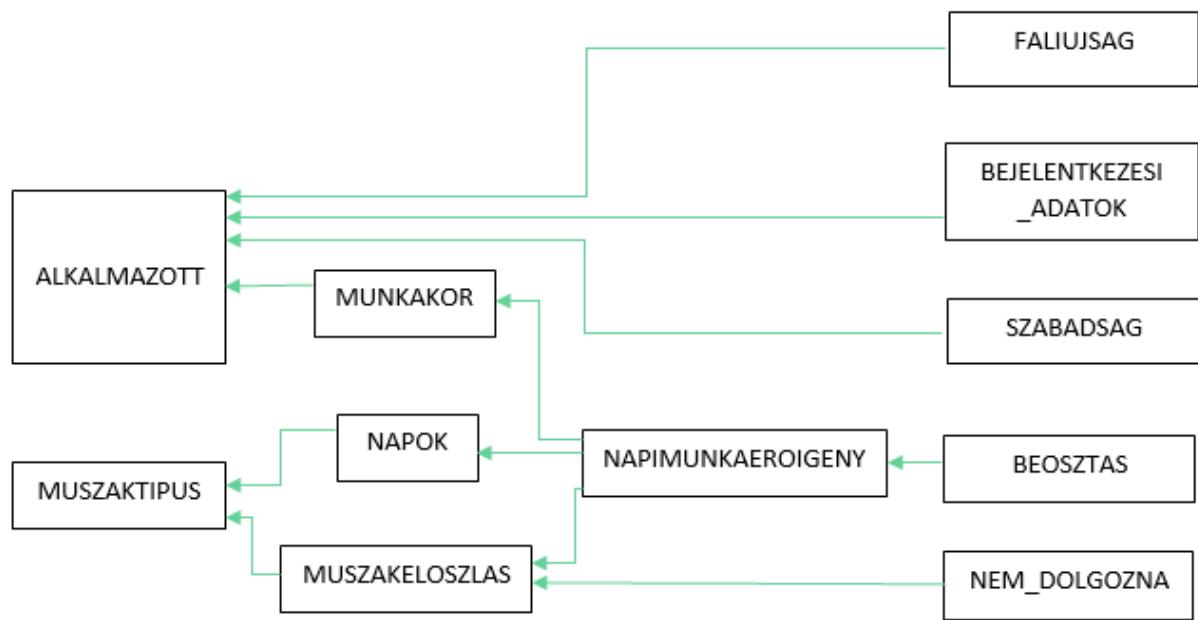
4.3.3 Szerkezet

ALKALMAZOTT	{ dolgozoi_azon, munkakor, adoazonosito, taj, elerhetoseg, email, heti_oraszam, nev, lakcim, szuletesi_datum, munkaviszony_kezdete, munkaviszony_vege }
MUNKAKOR	{ megnevezes, leiras ,munkafonok }
BEJELENTKEZESI_ADATOK	{ user_login, password, email }
FALIUJSAG	{ faliu_azonosito, dolgozoi_azon, mikor, cim, tartalom}
MUSZAKTIPOS	{ tipus, leiras }
MUSZAKELOSZLAS	{ muszakelo_azon, muszaktipus, muszakszam, oratol, oraig }

NAPIMUNKAEROIGENY	{ napim_azonosito, datum, muszakelo_azonosito, munkakör, db }
NAPOK	{ nap, muszaktipus, allapot }
BEOSZTAS	{ beo_azonosito, napim_azonosito, alkalmazott }
NEM_DOLGOZNA	{ nemdolgozna_azon, muszakelo_azon, alkalmazott, dátum }
SZABADSAG	{ szabadsag_azonosito, alkalmazott, tól, ig, szabadságtipus }
* Jelmagyarázat: TÁBLA	{ elsődleges_kulcs, külső_kulcs, egyedi kulcs }

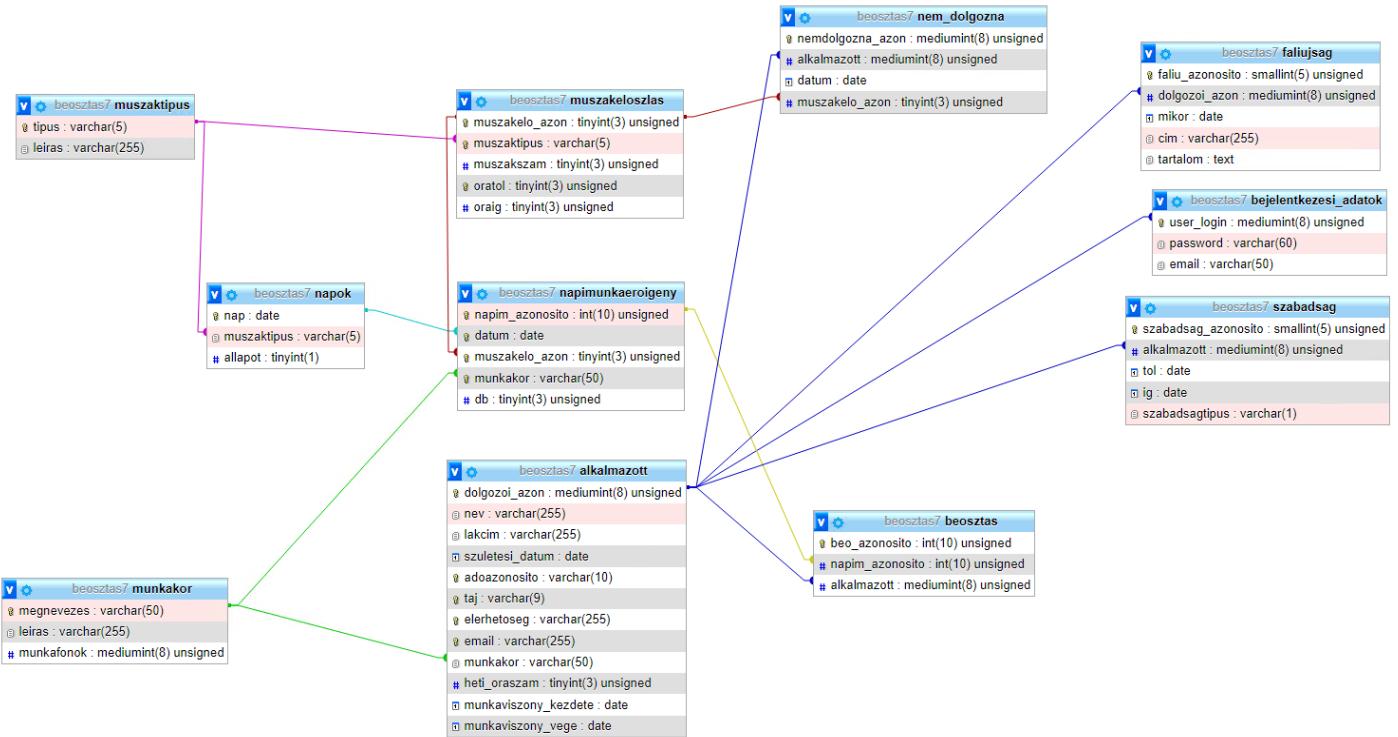
4.3.4 Egyed-kapcsolat diagram

Egyed-kapcsolat diagram



26. ábra - Egyed-kapcsolat diagram

4.3.5 Kapcsolati ábra



27. ábra - Kapcsolati ábra

4.3.6 Táblák szerkezete a tábla szintű megszorításokkal

ALKALMAZOTT			
Mező	Adattípus	Szerep	Korlátozás
dolgozói_azon	int	K	-
nev	varchar(255)		-
munkakor	varchar(50)	Külső Kulcs (MUNKAKOR)	-
adoazonosito	varchar(10)	Egyedi Kulcs	-
taj	varchar(9)	Egyedi Kulcs	-
elerhetoseg	varchar(255)	Egyedi Kulcs	-
email	varchar(255)	Egyedi Kulcs	-
heti_oraszam	tinyint		20 or 30 or 40
lakcim	varchar(255)		-
szuletesi_datum	date		now()-dátum>=14

munkaviszony_kezdete	date		-
munkaviszony_vege	date		dátum>dátum(munkaviszony_kezdete)

MUNKAKOR

Mező	Adattípus	Szerep	Korlátozás
megnevezes	varchar(50)	Kulcs	-
leiras	varchar(255)		-
munkafonok	mediumint		alapértelmezett érték: 0

BEJELENTKEZSI_ADATOK

Mező	Adattípus	Szerep	Korlátozás
user_login	mediumint	Kulcs, Külső Kulcs (ALKALMAZOTT)	-
password	varchar(60)		-
email	varchar(50)		

FALIUJSAG

Mező	Adattípus	Szerep	Korlátozás
faliu_azonosito	smallint	Kulcs	-
dolgozoi_azon	int	Külső Kulcs (ALKALMAZOTT)	-
mikor	date		-
cim	varchar(255)		-
tartalom	varchar(max)		-

MUSZAKTIPOS

Mező	Adattípus	Szerep	Korlátozás
tipus	varchar(5)	Kulcs	-
leiras	varchar(255)		-

MUSZAKELOSZLAS

Mező	Adattípus	Szerep	Korlátozás
muszakelo_azon	tinyint	Kulcs	
muszaktipus	varchar(5)	Külső Kulcs (MUSZAKTIPOS)	-
muszakszam	tinyint		-
oratol	tinyint		0<=oraig<=23
oraig	tinyint		0<=oratol<=23

NAPIMUNKAEROIGENY			
Mező	Adattípus	Szerep	Korlátozás
napim_azonosito	int	Kulcs	
dátum	date	Külső Kulcs (NAPOK)	datepart(week, dátum) == week()+1
muszakelo_azon	tinyint	Külső Kulcs (MUSZAKELOSZLAS)	-
munkakor	varchar(50)	Külső Kulcs (MUNKAKOR)	-
db	tinyint		alapértelmezett érték: 0
NAPOK			
Mező	Adattípus	Szerep	Korlátozás
nap	date	Kulcs	-
muszaktipus	varchar(5)	Külső Kulcs (MUSZAKTIPUS)	-
allapot	bit	alapért.0	-
BEOSZTAS			
Mező	Adattípus	Szerep	Korlátozás
beo_azonosito	int	Kulcs	
napim_azonosito	int	Külső Kulcs (NAPIMUNKAEROIGEN Y)	
alkalmazott	mediumint	Külső Kulcs (ALKALMAZOTT)	-
NEM DOLGOZNA			
Mező	Adattípus	Szerep	Korlátozás
nemdolgozna_azon	mediumint	Kulcs	
muszakelo_azon	tinyint	Külső Kulcs (MUSZAKEOLSZLAS)	
alkalmazott	mediumint	Külső Kulcs (ALKALMAZOTT)	-
datum	date		dátum> now()
SZABADSAG			
Mező	Adattípus	Szerep	Korlátozás
szabadsag_azonosito	smallint	Kulcs	

alkalmazott	int	Külső Kulcs (ALKALMAZOTT)	-
tol	date		dátum> now()
ig	date		-
szabadsagtipus	varchar(1)		F or N or B

4.4 Backend

4.4.1 Modellek és Controllerek

4.4.1.1 Model

A Model feladata, hogy egységebbe zárja a program aktuális állapotát, interakciót valósítson meg az adatbázissal (például lekérje egy szűrésből származó adatokat), adatokat konvertál megfelelő adatszerkezetbe, adatokat dolgoz fel az állapothoz illeszkedő formátumba, kommunikál a controllerrel és válaszol aktuális állapot kéréseire (Adatbázissal való kommunikáció.)

- \app\Models\Alkalmazott.php
- \app\Models\BejelentkezesiAdatok.php
- \app\Models\Beosztas.php
- \app\Models\Faliujsag.php
- \app\Models\Munkakor.php
- \app\Models\MuszakEloszlas.php
- \app\Models\MuszakTipus.php
- \app\Models\NapiMunkaeroIgeny.php
- \app\Models\Napok.php
- \app\Models\NemDolgozna.php
- \app\Models\Szabadsag.php

4.4.1.2 View

A View a felhasználó számára megjelenő felület, ami a Controller-től érkező dinamikus adatokat jeleníti meg, küldi el, a felhasználó utasításait és kéréseit továbbítja a Controller felé, illetve megjeleníti a Model adatait, és azok változásaira reagál. (Adatmegjelenítés)

- \resources\views\admin
- \resources\views\login
- \resources\views\managermenu
- \resources\views\usermenu
- \resources\views\welcome.blade.php

4.4.1.3 Controller

A Controller határozza meg az app viselkedését, módosítja a nézetet a megfelelő válaszok alapján. Kapcsolatot biztosít a Model és View között. (Összeköti a Modelt és a Viewt.)

- Beosztas\app\Http\Controllers\AlkalmazottController.php
- Beosztas\app\Http\Controllers\BejelentkezesiAdatokController.php
- Beosztas\app\Http\Controllers\BeosztasController.php
- Beosztas\app\Http\Controllers\Controller.php
- Beosztas\app\Http\Controllers\FaliujsgagController.php
- Beosztas\app\Http\Controllers\HitelesitesController.php
- Beosztas\app\Http\Controllers\JelszoVisszaAllitasController.php
- Beosztas\app\Http\Controllers\MunkakorController.php
- Beosztas\app\Http\Controllers\MuszakEloszlasController.php
- Beosztas\app\Http\Controllers\MuszakTipusController.php
- Beosztas\app\Http\Controllers\NapiMunkaeroIgenyController.php
- Beosztas\app\Http\Controllers\NapokController.php
- Beosztas\app\Http\Controllers\NemDolgoznaController.php
- Beosztas\app\Http\Controllers\StatisztikaController.php
- Beosztas\app\Http\Controllers\SzabadsagController.php

4.4.2 API végpontok

Kérés Típusa	URI	Leírás
GET	/api/alkalmazott/search	alkalmazottak összes adatai között keres
GET	/api/alkalmazottak	összes alkalmazott összes adatát adja vissza
GET	/api/alkalmazott/{id}	adott azonosítójú alkalmazott összes adatát adja vissza
PUT	/api/alkalmazott/{id}	adott azonosítójú alkalmazott adatait frissíti
POST	/api/alkalmazott	új alkalmazottat vesz fel
DELETE	/api/alkalmazott/{id}	adott azonosítójú alkalmazottat töröl
GET	/api/alkalmazott/sort	rendezi az alkalmazottakat adott sor alapján, pl név szerinti rendezés csökkenő sorrendben: /api/alkalmazott/sort?_sort=nev&_order=desc
GET	/api/muszakeloszlasok	összes műszakeloszlás összes adatát adja vissza
GET	/api/muszakeloszlas/{id}	adott azonosítójú műszakeloszlás összes adatát adja vissza
PUT	/api/muszakeloszlas/{id}	adott azonosítójú műszakeloszlás adatit frissíti
POST	/api/muszakeloszlas	új műszakeloszlást vesz fel
DELETE	/api/muszakeloszlas/{id}	adott azonosítójú műszakeloszlást töröl
GET	/api/muszaktipusok	összes műszaktípus összes adatát adja vissza
GET	/api/muszaktipus/{tipus}	adott típusú (pl.: A) műszaktípus összes adatát adja vissza
PUT	/api/muszaktipus/{tipus}	adott típusú műszaktípus adatait frissíti
POST	/api/muszaktipus	új műszaktípust vesz fel
DELETE	/api/muszaktipus/{tipus}	adott típusú műszaktípust töröl
GET	/api/beosztasok/expand	összes beosztás összes adatát adja vissza, kiegészítve a hozzá tartozó alkalmazott és napi munkaerőigény összes adatával
GET	/api/beosztas/{id}/expand	adott azonosítójú beosztás összes adatát adja vissza, kiegészítve a hozzá tartozó alkalmazott és napi munkaerőigény összes adatával
GET	/api/beosztasok	összes beosztás összes adatát adja vissza
GET	/api/beosztas/{id}	adott azonosítójú beosztás összes adatát adja vissza
PUT	/api/beosztas/{id}	adott azonosítójú beosztás adatait frissíti
POST	/api/beosztas	új beosztást vesz fel
DELETE	/api/beosztas/{id}	adott azonosítójú beosztást töröl

Kérés Típusa	URI	Leírás
GET	/api/faliujsagok	összes faliújság összes adatát adja vissza
GET	/api/faliujsag/{id}	adott azonosítójú faliújság összes adatát adja vissza
PUT	/api/faliujsag/{id}	adott azonosítójú faliújság adatait frissíti
POST	/api/faliujsag	új faliújságot vesz fel
DELETE	/api/faliujsag/{id}	adott azonosítójú faliújságot töröl
GET	/api/munkakorok	összes munkakör összes adatát adja vissza
GET	/api/munkakor/{megnevezes}	adott megnevezésű (pl.: Szakács) munkakör összes adatát adja vissza
PUT	/api/munkakor/{megnevezes}	adott megnevezésű munkakör adatait frissíti
POST	/api/munkakor	új munkakört vesz fel
DELETE	/api/munkakor/{megnevezes}	adott megnevezésű munkakört töröl
GET	/api/napimunkaeroigenyek/expand	összes napi munkaerőigény összes adatát adja vissza kiegészítve a hozzá tartozó műszakeloszlás összes adatával
GET	/api/napimunkaeroigenyek	összes napi munkaerőigény összes adatát adja vissza
GET	/api/napimunkaeroigeny/{id}	adott azonosítójú napi munkaerőigény összes adatát adja vissza
PUT	/api/napimunkaeroigeny/{id}	adott azonosítójú napi munkaerőigény adatait frissíti
POST	/api/napimunkaeroigeny	új napi munkaerőigényt vesz fel
DELETE	/api/napimunkaeroigeny/{id}	adott azonosítójú napi munkaerőigényt töröl
GET	/api/napokossz	összes nap összes adatát adja vissza
GET	/api/napok/{nap}	adott dátumú (pl.: 2022-01-10) nap összes adatát adja vissza
PUT	/api/napok/{nap}	adott dátumú nap adatait frissíti
POST	/api/napok	új napot vesz fel
DELETE	/api/napok/{nap}	adott dátumú napot töröl
GET	/api/nemdolgoznaossz/expand	összes nemdolgozna bejegyzés összes adatát adja vissza, kiegészítve a hozzá tartozó műszakeloszlás összes adatával
GET	/api/nemdolgoznaossz	összes nemdolgozna bejegyzés összes adatát adja vissza
GET	/api/nemdolgozna/{id}	adott azonosítójú nemdolgozna bejegyzés összes adatát adja vissza
PUT	/api/nemdolgozna/{id}	adott azonosítójú nemdolgozna bejegyzés adatait frissíti

Kérés típusa	URI	Leírás
POST	/api/nemdolgozna	új nemdolgozna bejegyzést vesz fel
DELETE	/api/nemdolgozna/{id}	adott azonosítójú nemdolgozna bejegyzést töröl
GET	/api/szabadsagok/expand	összes szabadság bejegyzés összes adatát adja vissza, kiegészítve a hozzá tartozó alkalmazott összes adatával
GET	/api/szabadsagok	összes szabadság bejegyzés összes adatát adja vissza
PUT	/api/szabadsag/{id}	adott azonosítójú szabadság bejegyzés adatait frissíti
POST	/api/szabadsag	új szabadság bejegyzést vesz fel
DELETE	/api/szabadsag/{id}	adott azonosítójú szabadság bejegyzést töröl
GET	/api/bejelentkezesiadatok	összes bejelentkezési-adat összes adatát adja vissza
GET	/api/bejelentkezesiadat/{user_login}	adott azonosítójú (pl.: 40001) bejelentkezési-adat összes adatát adja vissza
PUT	/api/bejelentkezesiadat/{user_login}	adott azonosítójú bejelentkezési-adat adatait frissíti
POST	/api/bejelentkezesiadat	új bejelentkezési-adatot vesz fel
DELETE	/api/bejelentkezesiadat/{user_login}	adott azonosítójú bejelentkezési-adatot töröl
GET	/login	bejelentkezési oldalra irányít
GET	/logout	kijelentkeztet
GET	/loggeduser	a bejelentkezett felhasználó azonosítóját adja vissza
POST	/authenticate	bejelentkeztetést végzi
POST	/change	jelszó módosítást végez
POST	/elfelejtettjelszo	jelszó változtató linket küld
POST	/reset-password	jelszó módosítást végez elfelejtett jelszó esetében
GET	/api/aktualishet/expand	aktuális hétfő napjaihoz tartozó napok bejegyzést ad vissza kiegészítve a hozzá tartozó műszakeloszlás összes adatával
GET	/api/munkakorstat	munkakörökhöz tartozó alkalmazottak számát adja vissza
GET	/api/hetioraszamstat	heti óraszámokra bontva adja vissza az alkalmazottak számát
GET	/api/szabadsagstat	az alkalmazott nevét, és a szabadság mettől meddig dátumját adja vissza
GET	/api/dolgozottnapok	az alkalmazott nevét, a hónap számát és a hónapban dolgozott napok számát adja vissza

Kérés Típusa	URI	Leírás
GET	/api/napiposzt	azokat a faliújság bejegyzéseket adja vissza, amik az aktuális napon lettek felvéve
GET	/api/szabadsagkerok	azokat az alkalmazottakat adja vissza akiknek a szabadság kérelme nincs még jóváhagyva
GET	/api/jovohet	az aktuális héthez képest következő hétre bejegyzett napok adatait adja vissza
GET	/api/aktualishet	az aktuális héthez bejegyzett napok adatait adja vissza
GET	/api/alkalmazhatoak	a nem szabadságon lévő alkalmazottak összes adatát adja vissza
GET	/api/jovohet-nmi	az aktuális héthez képest következő hétre bejegyzett napi munkaerőigény összes adatát adja vissza

4.4.3 Autentikáció/Hitelesítés

Mivel az oldalt teljes egészében saját kódóból építettük fel, illetve single-page tehát az adott oldalhoz tartozó összes funkció ugyanazon az oldalon érhető el, újra töltés vagy átirányítás nélkül, ezért a könnyebb testreszabhatóság érdekében, nem a Laravel Breeze starter kit-jét használtuk, hanem külön felépítettük. A hitelesítéshez és jelszó visszaállításhoz szükséges két kontrollert a Beosztas\app\Http\Controllers\Auth mappában találjuk, HitelesitesController.php és JelszoVisszaAllitasController.php néven. Az újra töltés elkerülése érdekében minden hibánál úgynevezett ValidationException-t dob a kontroller, amit egy fetch ajax kéréssel kezelünk le és írjuk ki a hibát a felhasználónak. Az alkalmazást felkészítettük „brute-force” támadások ellen is, a Laravel beépített RateLimiter segítségével, 3 hibás bejelentkezési adat megadása után 1 percig figyelmen kívül hagyja a próbálkozásokat.

4.5 Frontend

4.5.1 Javascript és jQuery

A beosztáskészítő rendszerünkben a jQuery könyvtárat használtuk a kliens oldali kódok elkészítéséhez. Egy olyan Javascript könyvtár, amely segítségével egyszerűbben és rövidebben valósíthatunk meg Javascript utasításokat, mint például HTML elemek kiválasztása, manipulálása, események kezelés, AJAX kérések küldése. Az applikációnk egy single-page alkalmazás, azaz egy adott felhasználói felületen csak az aktív oldalrészét

jelenítjük meg, anélkül, hogy az oldalt újra töltenénk vagy egy másik oldalra irányítanánk a felhasználót. A felhasználói jogosultságok miatt több oldalt hoztunk létre. Ezek oldalak az ügymenet miatt fontosak:

•Bejelentkezés	resources/views/login/login.php
•Elfelejtett jelszó	resources/views/login/forgetpassword.php
•Jelszóvisszaállítás	resources/views/login/resetpassword.php
•Szerepkörválasztó	resources/views/welcome.blade.php
•Felhasználóoldal	resources/views/usermenu/usermenu.php
•Manager oldal	resources/views/managermenu/managermenu.php
•Admin oldal	resources/views/admin/admin.php

A felületek, részletessége miatt ezek az aloldalak több komponensekből állnak, ezért menüpontonként feldaraboltuk az összes oldalt, és a hozzájuk tartozókomponensek akkor renderelnek újra, amint változás következik be az adott elemet illetően.

A kiválasztott oldal első betöltődésekor minden feldarabolt részről készítünk egy másolatot, amit a jQuery clone() metódusával valósítunk meg. Lehetővé teszi, hogy egy mély másolatot készítsünk egy általunk tetszőlegesen kiválasztott tárolóról, elemről a hozzátartozógyerek elemekkel együtt. Amint a felhasználó kiválaszt egy menüpontot, kiürítjük az oldal fő tárolóját és a kiválasztott elemet töltjük be a helyére.

Az előbb említett aloldalakhoz készítettünk egy-egy script fájlt, ami az aloldal funkcionalitását valósítja meg.

•Manager oldal	public/js/managermenu/manager.js
•Felhasználó oldal	public/js/user/user.js
•Admin oldal	public/js/admin/admin.js

A Manager oldal sokrétűsége miatt szükségünk volt egy Oldalkezelő osztály létrehozására, ami az oldaldarabok mentését valósítja meg, és a menüpontokhoz tartozókattintás eseményeket hozza létre. Ennek az oldalnak a funkcionalitása és az oldal töltési sebessége miatt választottuk a *single-page* alkalmazást, mert a szerver-oldali kommunikációt, kéréseket aszinkron hívásokkal valósítjuk meg. Emiatt, ha minden funkcionalitás és kérés az oldal betöltődésekor futna le, az oldal hosszú betöltését eredményezné, viszont ezek a kérések most kizárálag akkor futnak le, ha az általunk

kívánt menüpontra kattintunk, ezzel csökkentve a töltési időt és fokozva a felhasználhatóságát az oldalnak.

A kérések indítására a jQuery \$.ajax() függvényét használjuk. Az oldalunk négy HTTP kérést tud megvalósítani, ennek a metódusnak a használatával. Létrehoztunk egy Ajax osztályt és ennek az osztálynak a tagfüggvényei képesek ezen feladatok ellátására:

- GET: ajaxApiGet(apivegpont,callback)
- POST: ajaxApiPost(apivegpont,data)
- DELETE: ajaxApiDelete(apivegpont,id)
- PUT: ajaxApiPut(apivegpont,id,data,callback)

A callback függvények, olyan függvények, amik paraméterként lesznek átadva egy másik függvénynek, metódusnak. Az ajaxApiGet() metódust például véve amennyiben sikeres volt a kérés és kapunk vissza adatokat, csak akkor fusson le a callback(), ha az összes adatunk megérkezett. Amíg nem kaptuk vissza az API-tól az adatokat, addig nem tudunk műveletet végezni velük. Ezért célszerű a használatuk az Ajax osztály tagfüggvényeiben.

Az oldal menüje és a komponensek animációinak a nagyrésze is a jQuery könyvtár használatával valósult meg. Fontos, hogy ezek az animációk akkor működnek megfelelően, ha a HTML elem, amin alkalmazni szeretnénk nem láthatóak. A használt animációkért felelős metódusok:

- fadeIn()
- fadeOut()
- slideDown()
- slideUp()
- hide()
- show()

4.5.2 ajax.js

Az ajax.js tartalom egyetlen osztályban merül ki, melynek célja, hogy HTTP kéréseket küldhessünk. Egy Ajax(token) példányosításához át kell adnunk a konstruktorának egy tokent, ami megakadályozza az olyan támadásokat, ahol a felhasználó nevében küld parancsokat a támadó. Az osztály tagfüggvényei:

- *ajaxApiGet(apivegpont, callback)* – HTTP GET kérést tudunk küldeni egy adott végpontnak. Paraméterek:

apivegpont: Az URL cím ahova a kérést küldjük.

callback: Olyan függvény, ami csak akkor fut le, ha az aszinkron kérés befejeződött.

Ez a függvény felhasználhatja a válaszból visszakapott adatokat.

- *ajaxApiDelete(apivegpont,id)* – HTTP DELETE kerést tudunk küldeni egy adott végpontnak. Paraméterek:

apivegpont: Az URL cím ahova a kérést küldjük.

id: Törölni kívánt adat azonosítója.

- *ajaxApiPut(apivegpont,id,data)* – HTTP PUT kérést tudunk küldeni egy adott végpontnak. Paraméterek:

apivegpont: Az URL cím, ahova a kérést küldjük.

id: Módosítani kívánt adat azonosítója.

data: Az adat amire módosítani szeretnénk id alapján.

- *ajaxApiPut(apivegpont,id,data,callback)* – HTTP PUT kérést tudunk küldeni egy adott végpontnak. Az előző metódus túlterhelése. Paraméterek:

apivegpont: Az URL cím, ahova a kérést küldjük.

id: Módosítani kívánt adat azonosítója.

data: Az adat amire módosítani szeretnénk id alapján.

callback: Olyan függvény, ami csak akkor fut le, ha az aszinkron kérés befejeződött.

Ez a függvény felhasználhatja a válaszból visszakapott adatokat.

- *ajaxApiPost(apivegpont,data)* – HTTP POST kérést tudunk küldeni egy adott végpontnak

apivegpont: Az URL cím, ahova a kérést küldjük.

data: A rögzítendő adat.

4.5.3 Google Charts

A rendszerben használt diagrammokat, statisztikákat egy Javascript alapú könyvtár, a Google Chart használatával jelenítettük meg. Egyszerű használata miatt egy remek választás, ha adatokat szeretnénk megjeleníteni az oldalunkon. A diagrammok HTML5/SVG technológiával készülnek, biztosítva a böngészők közötti kompatibilitást (beleértve a régebbi IE verziókat is), valamint a hordozhatóságot iPhone, iPad és Android eszközökre is. Ennek eredményeként a felhasználóknak nem kell beépülő modulokkal, szoftverekkel bajlódniuk egyszerűen, ha van böngészőjük láthatják az oldalon használ sz diagrammokat.

- Beágyazzuk a weboldalunkba a Google Chart API-t.
- Betöltsük a kívánt Google Chart könyvtárat.
- Kiválasztjuk a diagramm típusát.
- Felsoroljuk a megjelenítendő adatokat.
- Beállítjuk milyen tulajdonságai legyenek a diagramnak (magasság, szélesség, szín stb.)
 - Létrehozunk egy `<div id="chart-helye"> </div>`-et, amiben megjelenítjük a diagramunkat.
 - Reszponzív alakítjuk, amire úgy van lehetőségünk, ha a Javascript `window.resize` eseménykor újra meghívjuk a diagramm rajzolásáért felelés metódusunkat.

4.5.4 Font Awesome

Az applikáció ikonjait a Font Awesome szabadon felhasználható ikonjaival valósítottuk meg. Nem minden ikon ingyenes, viszont hatalmas választék van azokat az ikonokat illetően, melyek közkedveltek a legtöbb weboldal design elkészítésekor. Ezek az ikonok kedvünk szerint alakíthatóak, színezhetőek, méretezhetőek CSS vagy inline formázások segítségével.

A legegyszerűbb módja a használatának, hogy egy csomagkezelővel feltelepítjük az ingyenes csomagot:

```
npm install --save @fortawesome/fontawesome-free
```

Az oldalunkra beágyazzuk, a szükséges css és script fájlokat, ezután minden ingyenes ikon elérhető lesz a projektünkben.

A fontawesome.com-on kedvünkre szűrhetünk és kereshetünk az ikonok között. Kiválasztva a preferált ikont, egy HTML tag osztályának értékül adjuk az ahhoz tartozó kódot. Például, ha egy HOME link mellé szeretnénk tenni egy házat, akkor a következőt kell tennünk:

```
<div><span class="fa fa-home"></span><a>Home</a></div>
```

A Font Awesome másik előnye, hogy nem csak native Javascripttel lehet használni, és a formázást sem csak CSS-sel tudjuk megoldani (SCSS, LESS). Kiválóan működik React, Vue, Angular, stb... használatával is. Erről többet megtudhatunk a Font Awesome hivatalos dokumentációjából.

4.5.5 CSS

Az oldal stílusát CSS (lépcsőzetes stíluslapok) segítségével valósítottuk meg. Hasonlóan a script fájlok esetén, minden oldalnak külön létrehoztunk egy-egy .css fájlt, amit beágyaztunk az oldal <head> tag-jébe. Az oldalon használt betűtípusok webbiztos betűtípusok, azaz minden platformon elérhetőek. Emellett az oldalon használt színeket css változókban tároljuk el, ezzel könnyítve a színek újra-felhasználhatóságát. A reszponzivitásért felelős töréspontokat is itt definiáltuk, valamint minden elem formázását az elemek megfelelő HTML osztályán keresztül állítottuk be.

•Bejelentkezés	public/css/login/login.css
•Elfelejtett jelszó	public/css/forgetpassword.css
•Jelszóvisszaállítás	public/css/resetpassword.css
•Felhasználó oldal	public/css/user/usermenu.css
•Manager oldal	public/css/managermenu/managermenu.css
•Admin oldal	public/css/admin/admin.css

4.5.6 user.js

A felhasználói oldal funkcionálisáért felelős javascript fájl. Metódusai:

- faliujsagUser(adatok) – Az oldal betöltésekor megjeleníti a faliújság posztjait-

A függvényen belül található Post osztályt használja a bejegyzések megjelenítéséhez. Az osztály fő eseménye egy kattintás esemény, amely megjeleníti a faliújság tartalmát. Paraméterei:

- adatok: A megjelenítendő posztok.
- class Post{
 constructor(szulo, adat, kep, alkalmazottAdatok)
}
 ■ Post paraméterei:
 szulo: Az a tároló, amiben a posztokat megszeretnénk jeleníteni.
 adat: Egy olyan objektum, amely a faliújság kulcs/érték párait.
 kep: Posztot készítő dolgozó képe.
 alkalmazottAdatok: Egy olyan objektum, amely a posztoló adatait tartalmazza.

- pagiJobbBal(szulo,tabla,sablon,elem,elemPerOldal) – Az oldalon lévő lapozható elemek létrehozását és elkészítését végzi. pl.: Faliújság
- newPost() – A bejelentkezett felhasználó új faliújság bejegyzését valósítja meg.
- profilAdatok() – Az oldal betöltésekor megjeleníteni a bejelentkezett felhasználó adatait, valamint minden egyes meghívásakor újra elvégzi ezt a műveletet.
- nemErekRaUser() – Létrehozza a naptárat, az információs ablakot és ezekhez az elemekhez tartozó eseményeket.
 - class Naptar{
 constructor(szulo, napok, honapok)
}

Megjeleníti az aktuális hónapot és példányosítja a Nap objektumokat.
Naptar paraméterei:

szulo: az a tároló, ahol megjelenítjük a naptárat.

napok: egy olyan tomb ami szövegként tartalmazza a napokat. pl.:
"H" "K" "SZ" "CS" "P" ...

honapok: egy olyan tomb ami szövegként tartalmazza a hónapokat.
pl.: "Január", "Február"

- class Nap{
 constructor(szulo, nap, napok, elem)
}

- Nap paraméterei:

szulo: az a tároló, ami tartalmazza a naptárat.

nap: egy szám, melyik naptári nap pl.: 10, 15, 31...

napok: egy olyan tomb ami szövegként tartalmazza a napokat. pl.: "H" "K"
"SZ" "CS" "P" ...

elem: az a tároló, ami az adott nap eleme lesz.

- userBeosztas() – Megjeleníti a bejelentkezett felhasználó heti beosztását.

4.5.7 manager.js

A manager oldal funkcionálisáért felelős javascript fájl.

- class Oldalkezelo{
 constructor(tarolo)
}

Megvalósítja a manager oldal single page viselkedését. Paraméterei:

tarolo: Az a főtároló ahova minden komponenst betoltunk. pl.: main, article, az
ujBeosztasPage(oldal) ennek a referenciáját használja.

- ujBeosztasPage(oldal) – Létrehozza a beosztáskészítő felületét, feltölti
komponensekkel, és a komponensekhez tartozó eseményeket hozzárendeli, az
oldalhoz. Paraméterei:
oldal: new Oldalkezelo() referenciaja.

- `bejelentkezettFelhasznalo(callback)` – Lekéri a bejelentkezett felhasználó adatait, és lehetővé teszi, hogy műveleteket végezzünk az adatokkal.
Paraméterei:

callback: egy olyan függvény, ami az adat lekérés után valósul meg.

- `udvozloUzenet(adatok)` – BejelentkezettFelhasznalo() callback függvénye, a felhasználó köszöntését valósítja meg. Paraméterei:

adatok: A bejelentkezett felhasználó.

- `munkakorok()` – Munkakörök komponenst betölti a fő tárolóba és a komponenshez tartozó funkcionálitást valósítja meg.
- `muszakok()` – Műszakok komponenst betölti a fő tárolóba és a komponenshez tartozó funkcionálitást valósítja meg.
- `muszakEloszlas()` – Műszakeloszlás komponenst betölti a fő tárolóba és a komponenshez tartozó funkcionálitást valósítja meg.
- `managerStatisztika()` – Statisztikák komponenst betölti a fő tárolóba a Google-chart api segítségével.
- `faliujsag()` – Faliújság komponenst betölti a fő tárolóba és a komponenshez tartozó funkcionálitást valósítja meg.
- `newPost()` – Új faliújság bejegyzését valósítja meg.
- `alkalmazottTabla()` – Alkalmazottak tábla komponenst betölti a fő tárolóba és a komponenshez tartozó funkcionálitást valósítja meg.
- `napiMin()` – Napi munkaerőigény komponenst betölti a fő tárolóba és a komponenshez tartozó funkcionálitást valósítja meg.

- ProfilAadatok(id) – Profiladatok komponenst betölti a fő tárolóba és a komponenshez tartozó funkcionálitást valósítja meg. Alkalmazottak tábla profil megtekintésekor hívódik meg. Paraméterei:

id: betöltendő alkalmazott azonosítója.

- pagination(szulo,tabla,sablon,elem,elemPerOldal) – Az oldalon lévő lapozható elemek létrehozását és elkészítését végzi. pl.: Faliújság. Paraméterei:

szulo: Az a tároló, ami tartalmazza a paginationt.

tabla: Az a tároló, amin a lapozást hajtjuk végre.

sablon: A nem lapozható elemek beazonosítására szolgál.

elem: az a tároló, ami a táblázat egyes elemeit határozza meg.

elemPerOldal: Az egy oldalon megjelenített elemeket határozza meg.

- beosztasMegtekinkes() – A beosztások komponenst betölti a fő tárolóba és a komponenshez tartozó funkcionálitást valósítja meg.

4.5.8 Osztalyok.js

Az egész projektben felhasznált osztályok állománya.

- class AlkalmazottTabla { constructor(szulo, adat, index) }
- class Munkakor { constructor(szulo, adat, ajax) }
- class MunkakorAlkalmazott { constructor(kep,adat,elem) }
- class MuszakEloszlas { constructor(szulo,adat,ajax) }
- class MuszakHozzaAdas { constructor(szulo, adat, ajax) }
- class Muszak { constructor(szulo, adat) }
- class Faliujsag { constructor(szulo, adat, ajax) }
- class NapiMin { constructor(szulo, adat, datum) }
- class Adminelemek{ constructor(szulo, adat, ajax) }

- class Alkalmazott extends Adminelemek { super(szulo, adat, ajax); //saját adattagok }
- class FaliujsagPost extends Adminelemek{ super(szulo, adat, ajax); //saját adattagok }

- class MunkakorA extends Adminelemek{ super(szulo, adat, ajax); //saját adattagok }
- class Bejelentkezes extends Adminelemek{ super(szulo, adat, ajax); //saját adattagok }
- class Muszaktipus extends Adminelemek{ super(szulo, adat, ajax); //saját adattagok }
- class Napimunkaeroigeny extends Adminelemek{ super(szulo, adat, ajax); //saját adattagok }
- class Napok extends Adminelemek{ super(szulo, adat, ajax); //saját adattagok }
- class Beosztas extends Adminelemek{ super(szulo, adat, ajax); //saját adattagok }
- class Szabadsag extends Adminelemek{ super(szulo, adat, ajax); //saját adattagok }
- class Nemdolgozna extends Adminelemek{ super(szulo, adat, ajax); //saját adattagok }

4.5.9 Oldalesemenyek.js

Az rendszerben használt menük viselkedésének eseményei, felhasználó preferáltságú funkciók, jelszavak és autentikáció kezelés.

- function sideNav() – Oldalsó menü megnyitásáért, bezárásáért felel.
- function changePass() – Jelszó megváltoztatása form megnyitásáért, bezárásáért felel.
- function compactMode() – Kompakt mód ki/be kapcsolása.
- function postAll() – Ellenőrzi mikor zárható be egy faliújság poszt.
- function dropDownMenus() – A főmenü legördülő menüjeit jeleníti meg/tünteti el.

4.6 Reszponzivitás

4.6.1 Mit jelent a reszponzív weboldal?

A reszponzív weboldalak tökéletes megjelenést biztosítanak minden eszközön a böngésző méretéhez igazodva. A reszponzív weboldalak a könnyű olvashatóság és egyszerű navigáció jellemző. A reszponzív design a legkényelmesebb és legteljesebb felhasználói élmény biztosítása érdekében elengedhetetlen. A felhasználók számára fontos, hogy a megkezdett munkát, folyamatot folytathassák mobilon, laptopon, ezen felül a Google SEO-ja is hátrébb sorolja azokat a weboldalakat, amik nincsenek mobilra optimalizálva, azaz nem mobilbarátak. A reszponzivitás alatt valójában nem egy programozási nyelvet vagy programot értünk, hanem egy tervezési és fejlesztési metódust.

4.6.2 Hogyan vágjunk bele a tervezésbe?

Telefon és tablet eszközök szempontjából fontos, hogy a tervezésbe az ujjaink méretét is bele kalkuláljuk, hiszen egy kisebb elemet, menüpontot lehet, nehezen tudnánk elérni és a felhasználói élmény szempontjából zavaró lehet, ha egy linkre háromszor, négyeszer is kattintanunk kell. Számolnunk kell azzal is, hogy az elemeink szélessége, magassága a kijelző méretéhez és magasságához is igazodjon, azaz reszponzív töréspontokat helyezzünk el az oldalon. Ajánlott figyelembe venni, hogy az eszköz melyik területét érjük el fizikailag a legkönyebben, például mobil eszközökön a bal alsó sarok, még táblagépeken a jobb felső sarok a legelérhetőbb. Ügyelnünk kell arra az esetre is, ha a felhasználó a kijelzője felén, negyedén használná csak a weboldalt, mert közben a képernyő másik felén másik felületet kezel.

Érdemes még törekedni arra az elvre is, hogy ne csak kis képernyőkben gondolkodjunk, hanem nagyobb kijelzők felé is tervezzünk. A tipográfia és a szövegek olvashatósága szintén elengedhetetlen feltétele a reszponzív felhasználói felület kialakításának. Érdemes a szülő elemhez relatív mértékegységeket használnunk a betűméretek megadásakor és az összefüggő szövegeket blokkokba törölve megjelenítenünk a felületen. Az oldal menüpöt praktikus úgy elkészíteni, hogy kis képernyőkön könnyen lenyithatóak legyenek és a szemünk automatikusan fókuszáljon a menüpontokra. Szövegeket is tagolhatjuk hasonló módon, és akkor választható, hogy

melyik szövegrészt szeretnénk megnyitni olvasásra. Ne feledkezzünk meg a képekről sem! A képek mérete és felbontása az oldal betöltési sebességét nagyban befolyásolja. A képek méretét minden ellenőrizzük le, mert lehet egy nagyfelbontású monitoron túlságosan dominánsok lehetnek.

4.6.3 Reszponzív kialakítás

A rendszer komponenseinek elrendezését grid technológiával oldottuk meg. minden komponens saját formázást kapott, többségük flex és grid tulajdonságai miatt szükségszerűen változik meg az oldal méretezésére. Az alábbi töréspontokat alkalmaztuk a rendszer kinézetének felépítésekor:

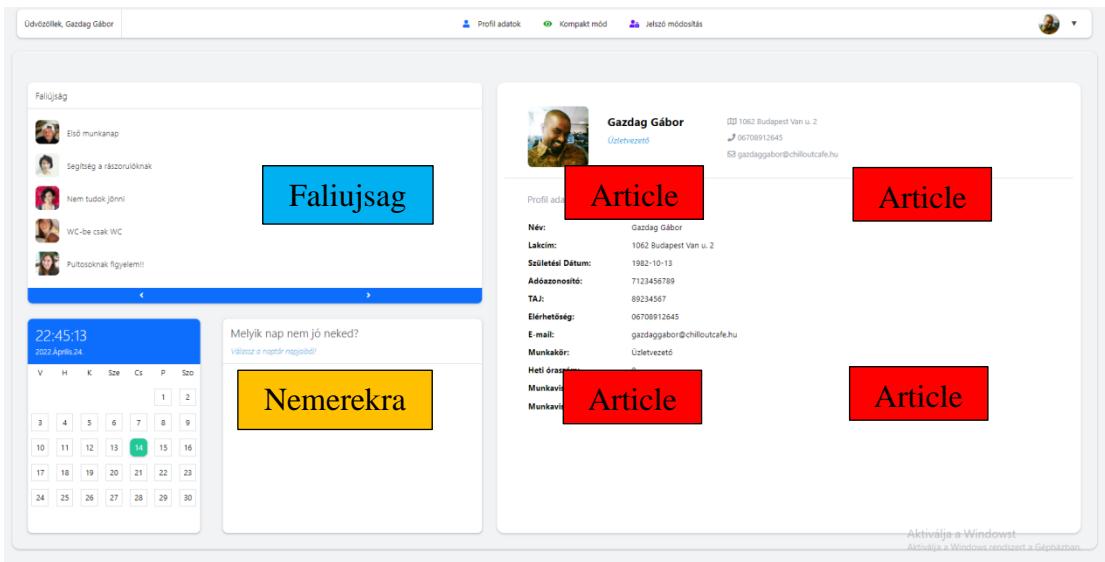
- Teljes képernyő
- 1200px
- 1024px
- 768px
- 480px

```
height: fit-content;
display: grid;
grid-template-areas:
"Faliujsag article article"
"Nemerekra article article";
gap:2rem;
```

28. ábra - User főtároló grid elrendezés 1300px felett

A user oldal főtárolójának (*container*) grid elrendezését a grid-template-areas kóddal valósítottuk meg, melynek során a már létrehozott html elemre hivatkozva

(annak tipusával, vagy id-jával) kellett megadnunk a főtárolóban elfoglalt helyét. A felület a grid felosztást figyelembe véve 1300px felett a felhasználó számára ebben a formában jelenik meg.



29. ábra - User oldal grid elrendezés 1300px felett

Mobil készülékeken szükséges volt, hogy az egyes tárolók ne horizontálisan, hanem vertikálisan jelenjenek meg annak érdekében, hogy a használó ne a bal-jobb irányú navigációt, hanem a telefonos platformokon megszokott, és kényelmesebb fel-le navigációt alkalmazza. Ezt megvalósítandó az oldal fő tárolóján belüli elemek elrendezését 1300px alatt megváltoztattuk, és az elemek egymás alá kerültek.

```

@media screen and (max-width:1300px) {
    .container{
        grid-template-areas:
        "Faliujsag"
        "article"
        "Nemerekra"
        "Beosztas";
    }
}

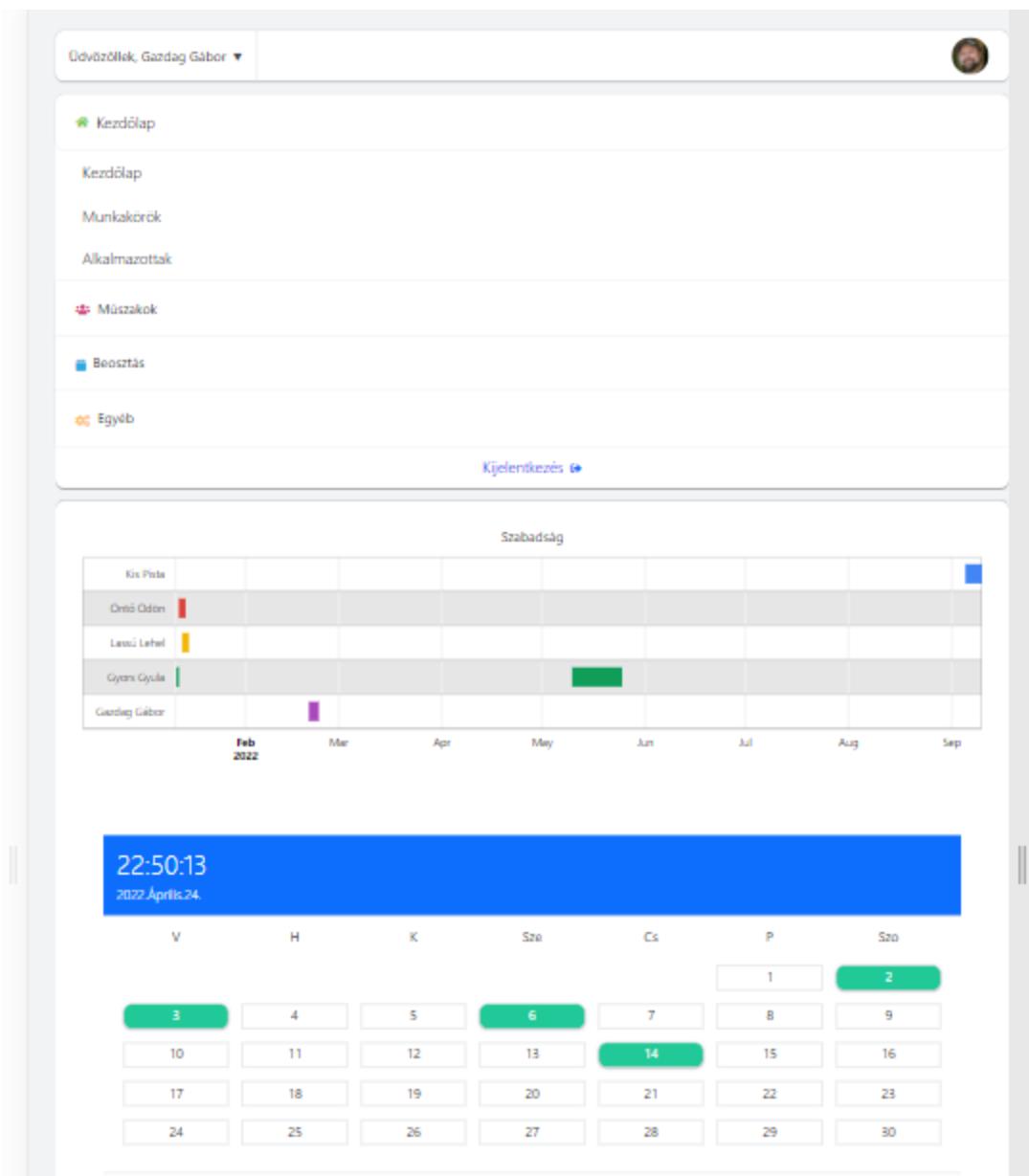
```

30. ábra - User főtároló grid elrendezés 1300px alatt

The screenshot displays a user profile page with a grid-based layout. At the top, there's a navigation bar with links for 'Ödvözöllek', 'Gázdag Gábor', 'Profil adatok', 'Kompakt mód', 'Jelszó módosítás', and a user icon. Below the navigation is a sidebar titled 'Faliújság' containing five items, each with a small profile picture and text: 'Első munkanap', 'Segítség a rászorulóknak', 'Nem tudok jönni', 'WC-be csak WC', and 'Pultosoknak figyelem!!'. To the right of the sidebar is a large blue box labeled 'Faliujsag'. The main content area features a profile picture of 'Gázdag Gábor' (labeled 'Üzletvezető'), contact information ('1062 Budapest Van u. 2', '06708912645', 'gazdaggabor@chilloutcafe.hu'), and a red box labeled 'Article'. Below this is a table of profile details: Név: Gázdag Gábor, Lakkim: 1062 Budapest Van u. 2, Születési Dátum: 1982-10-13, Adóazonosító: 7123456789, TAJ: 89234567, Elérhetőség: 06708912645, E-mail: gazdaggabor@chilloutcafe.hu, Munkakör: Üzletvezető, Heti óraszám: 0, Munkaviszony kezdete: 2021-03-15, and Munkaviszony Vége: -. At the bottom left is a calendar for April 2022, showing the date 22:44:39 and the 24th highlighted in green. To the right of the calendar is a yellow box labeled 'Nemerekra'.

31. ábra - User oldal grid elrendezés 1300px alatt

Mobil eszközökön a felhasználói élmény érdekében a nagyobb platformokon megjelenítendő menüt (*scrolldown / lenyíló*) átalakítottuk egy statikus navigációs menüre, ami állandóan elérhető a felhasználó számára. mindenépp fontosnak tartottuk, hogy a menü az oldal tetején helyezkedjen el az információs sáv alatt. Lehetőségük van a felhasználónak a menü megjelenítésének ki-bekapcsolására.



32. ábra - Mobilnézet menüsáv

5 Tesztelés

Login		
Teszteset	Várt eredmény	Kapott eredmény
Bejelentkezési kísérlet jó adatokkal.	Az oldal tovább irányít a szerepkörválasztóra, alkalmazott esetén az alkalmazott oldalra.	Várt eredménnyel azonos lett.
Bejelentkezési kísérlet hibás adatokkal.	Az oldal visszajelz a hibás próbálkozásról. Maradék próbálkozás száma X.	Várt eredménnyel azonos lett.
Bejelentkezési kísérlet csak email címmel.	Az oldal hibás bejelentkezési kísérletet érzékel, mert az email cím nem bejelentkező adat.	Várt eredménnyel azonos lett.
Bejelentkezési kísérlet elrontása többször.	Az oldal 1 perc kötelező várakozásra utasítja a felhasználót, nem engedve az új próbálkozást.	Várt eredménnyel azonos lett.
Elfelejtettek a jelszavad linkre kattintva.	Az oldal tovább irányít az /elfelejtett-jelszo oldalra, ahol folytathatjuk az ügymenetet.	Várt eredménnyel azonos lett.
Bejelentkezés alkalmazottként.	Az oldal tovább irányít a /usermenu aloldalra.	Várt eredménnyel azonos lett.
Bejelentkezés adminként.	Az oldal tovább irányít a szerepkör választóhoz.	Várt eredménnyel azonos lett.
Bejelentkezés managerként.	Az oldal tovább irányít a szerepkör választóhoz.	Várt eredménnyel azonos lett.

User oldal		
Teszteset	Várt eredmény	Kapott eredmény

<p>Bejelentkezés után az oldal betölti a megjelenítendő komponenseket,</p> <ul style="list-style-type: none"> • menübár • faliújság • nem érek rá (naptár) • profil adatok <p>és üdvözli a bejelentkezett felhasználót.</p>	<p>Az oldal sikeresen megjeleníti a komponenseket, az oldal használatra kész.</p>	<p>Várt eredménnyel azonos lett.</p>
<p>Menü bár használata.</p>	<p>A legördülő menüre kattintva a felhasználó menü elemei megjelennek. A Menü linkjeire (profil adatok, kompakt mód, jelszó módosítás) kattintva megjelennek a kívánt komponensek. A főmenü elemeire kattintva az oldal a container elembe betölti az aktív tartalmat.</p>	<p>Várt eredménnyel azonos lett.</p>
<p>Faliújság használata. (Oldalsó navigációs panel)</p>	<p>A faliújság miniatűr változata a nyilakra jól reagál, a posztok megtekinthetők, bezárhatóak.</p>	<p>Várt eredménnyel azonos lett.</p>
<p>Faliújság, új bejegyzés.</p>	<p>Megjelenik az űrlap és hozzáartozó segéd elemek. “Pipára” kattintva az új faliújság bejegyzés rögzítésre kerül, amennyiben minden adatot kitölöttünk. ”X”-re kattintva az űrlap bezárul. “Részletek” gombra kattintva megnyílik, bezárul a poszt dátuma és üzenete.</p>	<p>Várt eredménnyel azonos lett.</p>
<p>Faliújság menüpont, navigációs nyilak használata a főablakban.</p>	<p>“<” és “>” nyilakra kattintva a következő 5 post, visszamenőlegesen 5 post jelenik meg.</p>	<p>Javítandó, A gombok használata hibás, ha az első posztból egészen az utolsóig lapozunk majd újra szeretnénk kezdeni a folyamatot.</p>

Statisztika menüpont.	Üres statisztika oldal jelenik meg. (továbbfejlesztési lehetőség)	Várt eredménnyel azonos lett.
Beosztás menüpont.	Az alkalmazott beosztása töltődik be a fótárolóba.	Várt eredménnyel azonos lett.
Nem érek rá (naptár) kezelése.	A naptár elemeire (napok) kattintva a nap műszakeloszlásából válogathatunk, és határozzuk meg a nem preferált műszakjainkat. / Figyelmeztetés: a naphoz nincs műszakeloszlás beállítva	Várt eredménnyel azonos lett.

Manager oldal		
Teszteset	Várt eredmény	Kapott eredmény
Bejelentkezés után az oldal betölti a megjelenítendő komponenseket, <ul style="list-style-type: none"> • menübár • szabadságon • Beosztás készítő és üdvözli a bejelentkezett managert.	Az oldal sikeresen megjeleníti a komponenseket, az oldal használatra kész.	Várt eredménnyel azonos lett.
Menü bár használata.	A manager menü legördülő tartalmai, a menü gombjára kattintva, megnyílik és bezárhatóak. A menü linkjeire kattintva minden felület betölti az oldal aktív részét és megakadályozza az alapértelmezett esemény lefutását.	Várt eredménnyel azonos lett.

Kezdőlap menüpont.	Az oldal újratölt és behozza a manager menü kezdőállapotát.	Várt eredménnyel azonos lett.
Munkakörök menüpont.	“Új hozzáadás”-ra kattintva megjelenik a szükséges űrlap.”pipára” kattintva, minden “*”-gal jelölt input mező kitöltése esetén az új munkakör rögzítésre kerül. A “Részletek”-re kattintva a manager megtekintheti a munkakör dolgozóit. A dolgozói portrékra kattintva megtekintheti a dolgozó elérhetőségét.	Várt eredménnyel azonos lett.
Munkakörök “Részletek”-re való ismételt, gyors kattintás.	Az oldal egy kattintásig kéri le a munkakörhöz tartozó API eredményét.	Javítandó, több kérés indul el egyszerre.
Alkalmazottak menüpont.	Az alkalmazottak tábla navigációs részén a nyilak megfelelően működnek, a kereső mező reagál minden leütött billentyűre és visszaadja a helyes értékeket, “Ugrás ide” megfelelő oldalra lapoz. Adott sorra kattintva a “gyors módosítás” és “profil megtekintés” megjelenik.	Várt eredménnyel azonos lett.
Alkalmazottak menüpont gyors módosítás.	A “pipára” kattintva az adatok frissülnek az adatbázis megfelelő sorában és a felületen megváltozik a dolgozó adatsora.	Várt eredménnyel azonos lett.
Alkalmazottak menüpont profil megtekintés.	Az oldal betölti a dolgozó adatlapját, ahol minden adattal végezhetünk gyors módosítást.	Várt eredménnyel azonos lett.

Műszaktípus hozzáadása menüpont.	Az oldal megjeleníti az összes műszakot táblázatos formában, a hozzátartozó műszakfeloszlással. A felületen módosítható a műszak leírása. Megtekinthető a műszakfelosztása. Törölhetőek a műszakok.	Várt eredménnyel azonos lett.
Műszakeloszlás módosítása menüpont.	Az oldal betölti a főtárolóba az étterem műszaktípusait és hozzátartozó navigációs bárt. A nyilakra kattintva egyesével lépkedhetünk oda-vissza a műszakok között. Az aktív műszak, műszakeloszlását az adott sorra kattintva módosíthatjuk / törölhetjük.	Várt eredménnyel azonos lett.
Napi munkaerőigény menüpont.	Az oldal betölti a fő tárolóba az összes olyan napot, ahol már határozunk meg műszaktípust. Adott napra kattintva megjelennek az étterem munkakörei. Valamelyik munkakörre kattintva, megjelennek nap műszakjai és a műszakokhoz tartozó csúszska, amin beállítható a kívánt emberszám a nap aktuális intervallumához. A dátum alatt lévő tábla ikonokkal menthető/törölhető a nap munkaerőigénye.	Várt eredménnyel azonos lett.
Beosztás megtekintése menüpont.	Az oldal betölti a fő tárolóba az összes munkakört. Adott munkakörökre kattintva a hétközött rendelkező napjai jelennek meg, a kiválasztott munkakörre. A napokra kattintva megjelenik a nap beosztása.	Várt eredménnyel azonos lett.

Statisztika menüpont.	Az oldal a Google chart API-t használva 3 diagrammot tölt be a főtárolóba.	Várt eredménnyel azonos lett.
Faliújság menüpont.	Az oldal betölти a faliújság tárolóját a fő tárolóba. "Új bejegyzés"-re kattintva, megjelenik a szükséges űrlap. Az űrlap "pipa" elemre kattintva elküldhető az új bejegyzés, amennyiben a megfelelő input mezők kitöltésre kerültek. "X"-re kattintva bezárul a megnyitott űrlap. "Részletek"-re kattintva módosítható vagy törölhető az aktuális bejegyzés.	Várt eredménnyel azonos lett.
Jelszó módosítás	Az manager felületen végzett jelszó módosítás sikeres.	Javítandó, ehhez át kell lépni a user oldalra.
Kompakt mód	A betűtípus megváltozik, kisebb lesz.	Várt eredménnyel azonos lett.
Profil adatok	Az oldal betölти a manager adatait a főtárolóba.	Várt eredménnyel azonos lett.

Backend – UNIT tesztelés

```
PASS Tests\Feature\Http\Controllers\AlkalmazottControllerTest
✓ home
✓ get alkalmazottak
✓ get alkalmazott
✓ put alkalmazott
✓ post alkalmazott
✓ delete alkalmazott

Tests: 6 passed
Time: 2.68s
```

33. ábra - AlkalmazottController Teszt

```
PASS Tests\Feature\Http\Controllers\BejelentkezesiAdatokControllerTest
✓ get bejelentkezesiadatok
✓ get bejelentkezesiadat
✓ put bejelentkezesiadat
✓ delete bejelentkezesiadat

Tests: 4 passed
Time: 4.45s
```

34. ábra - BejelentkezesiAdatokController Teszt

```
PASS Tests\Feature\Http\Controllers\BeosztasControllerTest
✓ get beosztasok
✓ get beosztas
✓ put beosztas
✓ post beosztas
✓ delete beosztas

Tests: 5 passed
Time: 2.74s
```

35. ábra - BeosztasController Teszt

```
PASS Tests\Feature\Http\Controllers\FaliujsagControllerTest
✓ get faliujsagok
✓ get faliujsag
✓ put faliujsag
✓ post faliujsag
✓ delete faliujsag

Tests: 5 passed
Time: 2.58s
```

36. ábra - FaliujsagController Teszt

```
PASS Tests\Feature\Http\Controllers\MunkakorControllerTest
✓ get munkakorok
✓ get munkakor
✓ put munkakor
✓ post munkakor
✓ delete munkakor

Tests: 5 passed
Time: 2.58s
```

37. ábra - MunkakorController Teszt

```
PASS Tests\Feature\Http\Controllers\MuszakEloszlasControllerTest
✓ get muszakeloszlasok
✓ get muszakeloszlas
✓ put muszakeloszlas
✓ post muszakeloszlas
✓ delete muszakeloszlas

Tests: 5 passed
Time: 4.69s
```

38. ábra - MuszakEloszlasController Teszt

```
PASS Tests\Feature\Http\Controllers\MuszakTipusControllerTest
✓ get muszakeloszlasok
✓ get muszaktipus
✓ put muszaktipus
✓ post muszaktipus
✓ delete muszaktipus

Tests: 5 passed
Time: 5.32s
```

39. ábra - MuszakTipusController Teszt

```
PASS Tests\Feature\Http\Controllers\NapokControllerTest
✓ get napokossz
✓ get napok
✓ put napok
✓ post napok
✓ delete napok

Tests: 5 passed
Time: 2.66s
```

40. ábra - NapokController Teszt

```
PASS Tests\Feature\Http\Controllers\NemDolgoznaControllerTest
✓ get nemdolgoznaossz
✓ get nemdolgozna
✓ put nemdolgozna
✓ post nemdolgozna
✓ delete nemdolgozna
```

41. ábra - NemDolgoznaController Teszt

```
PASS Tests\Feature\Http\Controllers\SzabadsagControllerTest
✓ get szabadsagok
✓ get szabadsag
✓ put szabadsag
✓ post szabadsag
✓ delete szabadsag
```

Tests: 5 passed
Time: 2.555

42. ábra - SzabadsagController Teszt

6 Fejlesztési lehetőségek

A rendszer jelenleg megvalósítja azokat a funkciókat, amelyekkel elkészíthetünk egy általunk elképzelt órarendszerű beosztást, viszont pár kényelmi funkciót érdemes megemlíteni, amik javítanák a felhasználói élményt.

6.1.1 Aktuális ügymenet mentése

Ha szeretnénk készíteni egy beosztást az ügymenet lépései megkövetelik, hogy a hét napjaihoz hozzárendeljünk műszaktípust és adott naphoz rendeljünk hozzá munkaerőigényt is. Remek lehetőség lenne, ha gombnyomásra az előző heti beosztást betudnánk állítani az aktuális hétre, mert lehet teljesen megfelelő ugyanaz a beosztás, ami előző héten volt, annyi különbséggel, hogy egy-két napot esetleg órát szeretnénk csak módosítani. Így a felhasználónak nem kellene minden hét elején elkészítenie a beosztást, hanem sokkal gyorsabban és rugalmasabban tudna dolgozni.

6.1.2 Faliújságon kívül, személyes üzenetek

Egy belső levelező rendszer, chat alkalmazás megkönnyítené a kommunikációt a felületen.

6.1.3 A felhasználói felületet React-tel elkészíteni

A single-page alkalmazás megvalósítása React-tel sokkal gyorsabbá és egyszerűbbé tehetné a mostani ügymenetet, valamint fejlesztői szemszögből a kód rövidebb és könnyebben átlátható lehetne.

6.1.4 Éjszakai / Nappali mód

Manapság szinte alig látunk alkalmazásokat ez a funkció nélkül és az oldal szürke/fehér alapszínei kifejezetten zavarhatják a felhasználó szemét az esti órákban.

7 Irodalomjegyzék

Font Awesome – oldalon használt iconok	https://fontawesome.com/docs/web/setup/get-started
Szabadon felhasználható iconok	https://fontawesome.com/search?m=free&s=solid%2Cbrands
jQuery	https://api.jquery.com/
jQuery UI	https://api.jqueryui.com/
Google Charts – diagramm, statisztika	https://developers.google.com/chart/interactive/docs/quick_start
Kör diagramm	https://developers.google.com/chart/interactive/docs/gallery/piechart
Idővonal	https://developers.google.com/chart/interactive/docs/gallery/timeline
Oszlop diagramm	https://developers.google.com/chart/interactive/docs/gallery/columnchart
Google Fonts	https://fonts.google.com/
Normalize CSS	https://necolas.github.io/normalize.css/
jQuery.ajax()	https://api.jquery.com/jquery.ajax/
jQuery ajax() Esemény kezelők	https://api.jquery.com/category/ajax/global-ajax-event-handlers/
jQuery animaciók	https://api.jquery.com/category/effects/
jQuery DOM manipuláció	https://api.jquery.com/category/manipulation/
jQuery események	https://api.jquery.com/category/events/
Bootstrap	https://getbootstrap.com/docs/5.1/getting-started/introduction/
NPM	https://docs.npmjs.com/getting-started
Composer	https://getcomposer.org/doc/
Laravel és Artisan	https://laravel.com/docs/9.x https://laravel.com/docs/9.x/artisan
GitHub	https://docs.github.com/en/get-started
PHP	https://www.php.net/docs.php
JavaScript, ES6	https://developer.mozilla.org/en-US/docs/Web/JavaScript
Dribbble	https://dribbble.com/shots