



Vas Megyei Szakképzési Centrum  
Nádasy Tamás Technikum és Kollégium

# **PROJEKTFELADAT**

## **Reservelt**

**Csóka Csaba, Kővári Bálint**

**Konzulens:  
Balics Gábor**

**2023**

# Nyilatkozat

Alulírott, Csóka Csaba, Kővári Bálint kijelentem, hogy a ReserveIt című projektfeladat kidolgozása a saját munkám, abban csak a megjelölt forrásokat, és a megjelölt mértékben használtam fel, az idézés szabályainak megfelelően, a hivatkozások pontos megjelölésével.

Eredményeim saját munkán, számításokon, kutatáson, valós méréseken alapulnak, és a legjobb tudásom szerint hitelesek.

Csepreg, 2023. 04. 23.

---

hallgató

Csepreg, 2023. 04. 23.

---

hallgató

# Kivonat

## ReserveIt

Létrehoztunk egy olyan weboldalt, amelyen regisztráció után nyomon tudjuk követni a vállalkozások elérhetőségeit és a rendszerünk segítségével ők tudnak nekünk időpontokat adni, amelyet maga a vendég is lát a saját naptárán belül a többi időpontjával együtt. Vállalkozást bárki tud létrehozni, akinek van felhasználója és ahhoz hozzá is tud adni dolgozókat, akik szintén tudják kezelni a vállalkozás időpontjait.

# **Abstract**

## **ReserveIt**

In today's world, appointment booking plays an increasingly important role for service providers such as dentists and manicurists due to growing demand. It is becoming less and less common to access these services without a prior reservation. Appointments are usually tracked on paper, calendars, or notebooks, which can be disadvantageous in terms of post-editing or the lack of paper and pen. Furthermore, it can be problematic if the guest forgets or writes down their appointment incorrectly. Our program provides a solution to this problem.

# Tartalomjegyzék

1.	A program .....	9
1.1.	Probléma .....	9
1.2.	Alternatívák .....	9
2.	Felhasznált technológiák .....	10
2.1.	React .....	10
2.2.	CSS .....	10
2.3.	JavaScript.....	11
2.4.	NodeJS .....	11
2.5.	Express.JS .....	12
2.6.	Moment.js .....	12
2.7.	MySQL .....	13
2.8.	Jest .....	13
2.9.	Visual Studio Code .....	14
3.	ReserveIt.....	15
3.1.	Bejelentkező felület .....	15
3.1.1.	Regisztráció .....	15
3.1.2.	Bejelentkezés .....	15
3.1.3.	Elfelejtett jelszó .....	16
3.2.	Kezelőfelület.....	17
3.2.1.	Menü.....	17
3.2.2.	Időpontjaim.....	18
3.2.3.	Vállalkozások .....	20
3.2.4.	Saját vállalkozásaim .....	21
3.2.5.	Beállítások .....	23

3.3.	Vállalkozás kezelőfelület .....	24
3.3.1.	Menü.....	24
3.3.2.	Időpontok.....	25
3.3.3.	Dolgozók .....	26
3.3.4.	Beállítások .....	26
4.	Backend .....	28
4.1.	/addReservation .....	28
4.2.	/changePassword.....	28
4.3.	/checkNewPasswordKey .....	28
4.4.	/createOrganization .....	28
4.5.	/deleteOrganization .....	29
4.6.	/editReservation .....	29
4.7.	/forgottenPassword .....	29
4.8.	/getAllOrganizations .....	29
4.9.	/getBusinessInvites .....	29
4.10.	/getNotifications .....	29
4.11.	/getNotificationsCount .....	30
4.12.	/getOrganizations .....	30
4.13.	/getReservations .....	30
4.14.	/getUserFromEmail .....	30
4.15.	/getUserReservations.....	30
4.16.	/getWorkers .....	30
4.17.	/inviteWorker.....	30
4.18.	/isOrganizationAuthorized .....	31
4.19.	/loginUser.....	31
4.20.	/logout.....	31

4.21.	/recaptcha .....	31
4.22.	/registerUser .....	31
4.23.	/removeReservation.....	31
4.24.	/removeWorker.....	32
4.25.	/respondToInvite.....	32
4.26.	/updateOrganizationName.....	32
4.27.	/verifyAccount.....	32
4.28.	/verifyToken .....	32
5.	Adatbázis .....	33
5.1.	ReserveIt_Accounts .....	33
5.2.	ReserveIt_VerificationData.....	33
5.3.	ReserveIt_ForgottenPasswordData.....	33
5.4.	ReserveIt_Businesses.....	33
5.5.	ReserveIt_BusinessInvites .....	34
5.6.	ReserveIt_BusinessEmployees .....	34
5.7.	ReserveIt_Reservations .....	34
6.	Tesztelés.....	35
7.	Továbbfejlesztési lehetőség .....	36
7.1.	Havi fizetés .....	36
7.2.	Helyek.....	36
7.3.	Térkép .....	36
7.4.	Kuponok.....	36
7.5.	Dolgozói jogrendszer .....	36
7.6.	Statisztikák.....	36
8.	Források.....	38
9.	Ábrajegyzék.....	39

10.	Mellékletek.....	40
-----	------------------	----



### **1. A program**

#### **1.1. Probléma**

Napjainkban egyre nagyobb szerepet kap az időpont foglalás a szolgáltatók, mint például a fogászatok, manikűrösök munkája során. Egyre kevesebb esetben lehet foglalás nélkül ezeket a szolgáltatásokat elérni a növekvő igény miatt. A foglalásokat leggyakrabban papírlapon, naptárakban vagy jegyzetfüzetekben követik nyomon, amelyek hátrányosak lehetnek olyan szempontokból, mint az utólagos szerkesztés vagy maga a papír és toll hiánya. Ezen felül az is problémát okozhat, ha maga a vendég elfelejti vagy rosszul írja le magának az időpontját. A programunk erre ad megoldást

#### **1.2. Alternatívák**

Nyilvánvalóan vannak más ehhez hasonló megvalósítások, viszont legtöbbször ezeket magának a vállalkozásnak készítik vagy havi előfizetés formájában elérhetőek. A miénk jelenleg nem tartalmazza a fizetési lehetőséget, viszont meg van benne az a potenciál, hogy a főbb funkciók elérhetőek maradjanak és csak az igazán mellékes dolgokat kelljen megfizetni.

## 2. Felhasznált technológiák

### 2.1. React

React egy nyílt forráskódú JavaScript keretrendszer, amelyet a Facebook fejlesztett ki, és célja, hogy könnyebbé tegye a webalkalmazások fejlesztését. A React fő előnye az, hogy lehetővé teszi a webalkalmazások felépítését úgynevezett "komponensek" használatával, amelyek lehetővé teszik a fejlesztők számára, hogy a weboldalt különböző részeire osztva hozzanak létre újra felhasználható kódrészleteket. A React további előnye, hogy lehetővé teszi az egyszerű és hatékony adatkezelést az alkalmazásban, valamint, hogy támogatja a virtuális DOM (Document Object Model) koncepcióját, amely lehetővé teszi az alkalmazás számára, hogy hatékonyabban frissítse az adatokat és az állapotokat az oldalon. A React nagyon népszerű keretrendszer lett az elmúlt években, és sok nagy webalkalmazás, például a Facebook, Instagram és Airbnb is használja. Az egyszerűsége, hatékonysága és rugalmassága miatt sok fejlesztőnek a React az első választása webalkalmazások fejlesztésekor.

### 2.2. CSS

A CSS (Cascading Style Sheets) egy stílusleíró nyelv, amelyet a weboldalak megjelenítésének megváltoztatására és testre szabására használnak. A CSS lehetővé teszi a fejlesztők számára, hogy formázzák a HTML dokumentumokat, és megadják a weboldalak színeit, betűméretét, elrendezését, stílusát és egyéb vizuális jellemzőit. A CSS hatékonyan működik a HTML dokumentumokkal együtt, és lehetővé teszi a fejlesztők számára, hogy elválasszák a tartalmat és a formázást egymástól. Ez azt jelenti, hogy a HTML dokumentumok csak a tartalmat tartalmazzák, míg a CSS fájlok tartalmazzák a formázási szabályokat. Ez a megközelítés nagymértékben megkönnyíti a weboldalak testre szabását és karbantartását. A CSS-t széles körben használják a webfejlesztésben, és az egyik alapvető technológia, amely lehetővé teszi a modern weboldalak testre szabását és vizuális jellemzőinek megváltoztatását. A CSS használata lehetővé teszi a weboldalak széles körű testre szabását, és javítja a felhasználói élményt. A CSS egy nyílt szabvány és számos eszköz és keretrendszer áll rendelkezésre, amelyek segítenek a CSS-fájlok létrehozásában és karbantartásában.

### 2.3. JavaScript

JavaScript (rövidítve: JS) egy szkriptnyelv, amelyet általában weboldalakon használnak. A JavaScript lehetővé teszi a weboldalak interaktív és dinamikus funkcióinak hozzáadását, és kiegészíti azokat a statikus HTML és CSS kódokkal. A JavaScript programozási nyelv lehetővé teszi a fejlesztők számára, hogy weboldalakat, webes alkalmazásokat, szerveroldali programokat és még sok más alkalmazást fejlesszenek. A JavaScript olyan szintaktikai szabályokat tartalmaz, amelyek lehetővé teszik a változók, a függvények, az objektumok, az események és az állapotok kezelését. A JavaScript lehetővé teszi a weboldalak interaktív és dinamikus funkcióinak, például az animációknak, a játékoknak, az űrlapoknak, az adatkezelésnek, a dinamikus tartalomnak, az AJAX hívásoknak, az eseményeknek és az adatvalidációnak a fejlesztését. A JavaScript számos keretrendszert és könyvtárat kínál, amelyek megkönnyítik a fejlesztést, például a React, az Angular, a Vue, a jQuery és még sok más keretrendszert és könyvtárat. A JavaScript népszerűsége és sokoldalúsága miatt a fejlesztők körében az egyik legnépszerűbb programozási nyelvvé vált, amelyet széles körben alkalmaznak a webfejlesztés területén.

### 2.4. NodeJS

NodeJS egy nyílt forráskódú, szerveroldali JavaScript futtatókörnyezet, amely lehetővé teszi a fejlesztők számára, hogy JavaScript-ben írjanak és futtassanak kódot a szerveroldalon. A NodeJS az első olyan JavaScript futtatókörnyezet, amely lehetővé teszi a JavaScript használatát szerveroldali fejlesztéshez, és azóta széles körben használják a fejlesztők a webalkalmazások, az API-k és más szerveroldali alkalmazások fejlesztéséhez. A NodeJS a Google Chrome V8 JavaScript motorját használja, amely gyorsítja a kód futtatását és lehetővé teszi a hatékony memóriakezelést. A NodeJS-t aszinkron működése jellemzi, ami lehetővé teszi a fejlesztők számára, hogy hatékonyan kezeljék az adatokat és az eseményeket anélkül, hogy blokkolnák az alkalmazás futását. Emellett a NodeJS támogatja a moduláris fejlesztési módszert, amely lehetővé teszi a fejlesztők számára, hogy újra felhasználják és könnyen kezeljék a kódrészeket. A NodeJS-t széles körben használják a webalkalmazások és az API-k fejlesztéséhez, de használják például az adatbázis-kezeléshez, a szerveroldali szkriptek futtatásához, a fájlkezeléshez és még sok más feladathoz.

## **2.5. Express.JS**

ExpressJS egy nyílt forráskódú, szerveroldali NodeJS alapú webalkalmazás-keretrendszer, amely lehetővé teszi a fejlesztők számára, hogy gyorsan és egyszerűen építsenek webalkalmazásokat és API-kat. Az ExpressJS egy olyan kis keretrendszer, amely nagyon kevés megszorítást helyez a fejlesztőkre a felépítési módszerük és az alkalmazásuk architektúrájának kialakítása terén. Az ExpressJS egy nagyon rugalmas keretrendszer, amely lehetővé teszi a fejlesztők számára, hogy testre szabják az alkalmazásukat az egyedi igényeknek megfelelően. Az ExpressJS lehetővé teszi a fejlesztők számára, hogy könnyedén definiálják az útvonalakat, amelyeken keresztül az alkalmazás a kientől kapott HTTP kéréseket fogadja, és válaszoljon azokra. Az ExpressJS támogatja az aszinkron kód futtatását, amely lehetővé teszi a fejlesztők számára, hogy hatékonyan kezeljék az adatokat és az eseményeket az alkalmazásban. Az ExpressJS nagyon népszerű a NodeJS fejlesztői között, és számos nagy webalkalmazás és API használja, például a PayPal, a LinkedIn és a Uber. Az ExpressJS lehetővé teszi a fejlesztők számára, hogy hatékonyan és gyorsan fejlesszenek szerveroldali webalkalmazásokat és API-kat.

## **2.6. Moment.js**

A Moment.js egy nyílt forráskódú JavaScript könyvtár, amelyet a dátum- és időkezeléssel kapcsolatos feladatok megkönnyítésére használnak. A Moment.js lehetővé teszi a fejlesztők számára, hogy egyszerűen létrehozzanak, formázzanak és manipuláljanak dátumokat és időket a JavaScriptben. A Moment.js használata egyszerű és intuitív, és nagyban javítja az idővel kapcsolatos feladatok végrehajtásának hatékonyságát. A Moment.js számos funkciót kínál a dátum- és időkezeléssel kapcsolatban, például a dátumok és idők formázását, dátumok és idők összehasonlítását, dátumok és idők manipulálását, dátumok és idők átalakítását más formátumokba, és még sok más funkciót. A Moment.js támogatja a több időzónával kapcsolatos műveleteket is. A Moment.js használata különösen fontos, ha a fejlesztőknek dátum- és időkezeléssel kapcsolatos problémákat kell megoldaniuk a webalkalmazásaikban. A Moment.js lehetővé teszi a fejlesztők számára, hogy egyszerűen és hatékonyan kezeljék a dátumokat és időket, amelyek a webalkalmazásuk működéséhez szükségesek. A Moment.js-t számos nagy

webalkalmazás, például a GitHub és a LinkedIn is használja.

### 2.7. MySQL

MySQL egy nyílt forráskódú, relációs adatbázis-kezelő rendszer, amelyet a fejlesztők a strukturált adatok tárolására és kezelésére használnak. A MySQL-t az Oracle Corporation fejleszti és támogatja. A MySQL az egyik legnépszerűbb adatbázis-kezelő rendszer a világon, és széles körben használják a webalkalmazások, a serveroldali szoftverek és az adatfeldolgozási alkalmazások fejlesztéséhez. A MySQL használata egyszerű és hatékony. A MySQL-t SQL nyelven lehet programozni, amely lehetővé teszi az adatbázisban tárolt adatok lekérdezését, frissítését és kezelését. A MySQL támogatja a tranzakciókezelést, amely lehetővé teszi a fejlesztők számára, hogy biztonságosan és hatékonyan kezeljék az adatokat az adatbázisban. A MySQL használata különösen hatékony, ha nagy mennyiségű adatot kell tárolni és lekérdezni. A MySQL-t széles körben használják a webalkalmazások, a CMS-ek (tartalomkezelő rendszerek), a blogmotorok, a fórumok és más internetes alkalmazások fejlesztéséhez. Az adatbázis-kezelő rendszer lehetővé teszi az alkalmazások számára, hogy hatékonyan tárolják és kezeljék az adatokat, amelyekre az alkalmazás működése alapul. A MySQL egy megbízható és hatékony adatbázis-kezelő rendszer, amely széles körben használatos a világ számos területén.

### 2.8. Jest

A Jest egy JavaScript alapú tesztelő keretrendszer, amelyet a Facebook fejlesztett ki. A Jest célja, hogy egyszerű és hatékony tesztelést biztosítson a JavaScript alkalmazások számára. A keretrendszer számos beépített funkciót kínál, például aszinkron tesztek kezelését, az előkészítés és az utófeldolgozás funkcióit, valamint a kód lefedettségének ellenőrzését. A Jest az egységtesztelésre összpontosít, és számos környezeti funkciót kínál, amelyek lehetővé teszik a fejlesztők számára, hogy egyszerűen teszteljék az alkalmazásokat a különböző böngészőkben, illetve Node.js-ben. A keretrendszer támogatja a snapshot teszteket, amelyek lehetővé teszik, hogy a teszt keretrendszer automatikusan rögzítse a komponensek megjelenését, majd ellenőrizze, hogy a teszt során azok változatlanok maradtak-e.

## **2.9. Visual Studio Code**

A Visual Studio Code egy ingyenes és nyílt forráskódú kódszerkesztő és fejlesztői környezet, amelyet a Microsoft fejlesztett ki. A Visual Studio Code támogatja a több platformot, és számos programozási nyelvhez kínál fejlesztői eszközöket, például JavaScript, TypeScript, Python, C#, Java, PHP és még sok más nyelvhez. A Visual Studio Code felhasználóbarát felülettel rendelkezik, amely lehetővé teszi a fejlesztők számára, hogy könnyen és hatékonyan dolgozzanak a kódon. A Visual Studio Code kiterjesztéseket és plugineket kínál, amelyek lehetővé teszik a felhasználók számára, hogy testre szabják és bővítsék a funkcióit. A Visual Studio Code számos hasznos funkciót kínál a fejlesztők számára, például a kódkiemelést, a kódfelismerést, a szintaktikai ellenőrzést, a kódblokkok szerkesztését és törlését, a debugger használatát, a verziókezelést és még sok más funkciót. A Visual Studio Code nagyon népszerű a fejlesztők körében, mert nagyon hatékony, sokoldalú és bővíthető.

### 3. Reservelt

#### 3.1. Bejelentkező felület

##### 3.1.1. Regisztráció

Az oldal használata előtt regisztrálnunk kell egy felhasználót, amelyhez egy működő email címre lesz szükségünk. A regisztráció teljesen egyértelmű, meg kell adnunk a vezetéknévünket, keresztnévünket az email címünket és kétszer a kívánt jelszót. A megadott jelszónak tartalmaznia kell legalább egy számot, egy nagy betűt és nyolc karakterből kell állnia. Ha a rendszer mindent rendben talált, akkor értesítést fogunk kapni, hogy a regisztráció befejezéséhez meg kell erősítenünk az email címünket az arra küldött levélben lévő linkre kattintva. Ha a rendszer problémát talált, akkor az érintett mező alatti vonal vörösre vált és leírja a problémát.

1. ábra A regisztrációs felület

##### 3.1.2. Bejelentkezés

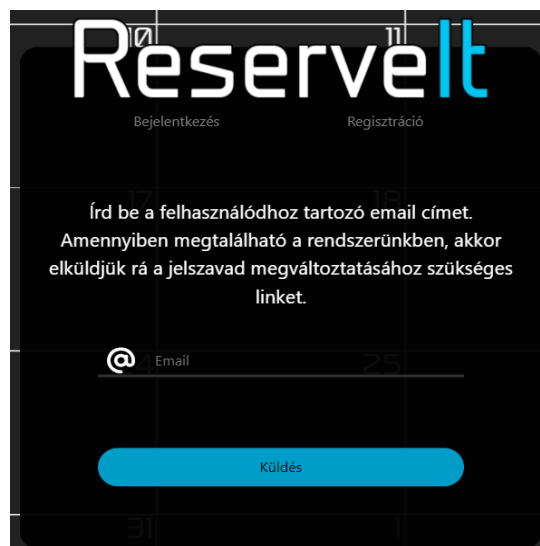
2. ábra A bejelentkező felület

Miután regisztráltunk egy felhasználót be tudunk jelentkezni az oldalra, ezt az email címünk és a jelszavunk megadásával tehetjük meg. A regisztrációhoz hasonlóan, ha valami hiba lép fel, akkor azt az érintett mező alatti vonal vörösre váltásával jelzi nekünk a rendszer. A bejelentkezés gombra kattintva szinte rögtön a vezérlőpanelban találjuk magunkat. A bejelentkező felületen lévő interakciókat reCAPTCHA v3 védelemmel láttuk el. Bejelentkezésnél nincs külön ellenőrzés arra, hogy már be van-e valaki

jelentkezve a profilunkba, így egyszerre több eszközön is tudjuk használni az oldalt kijelentkezés nélkül.

### 3.1.3. Elfelejtett jelszó

Ha elfelejtettük jelszavunkat, akkor egyszerűen rányomunk a bejelentkezésnél található „Elfelejtette a jelszavát?” felíratra és a panel rögtön átvált. Az újonnan megjelent szöveg leírja mit kell tennie a felhasználónak, avagy az email címe megadásával kapni fog rá egy levelet, amelyben egy link segítségével állíthat be magának egy teljesen új jelszót. Az új jelszóra ugyanúgy vonatkoznak a szabályok, vagyis 8 karakterből kell állnia és legalább 1 nagy betűt és számot kell tartalmaznia. Ha rámentünk a Megváltoztatás gombra már nincs más dolgunk, mint bejelentkezni az új jelszavunkal.



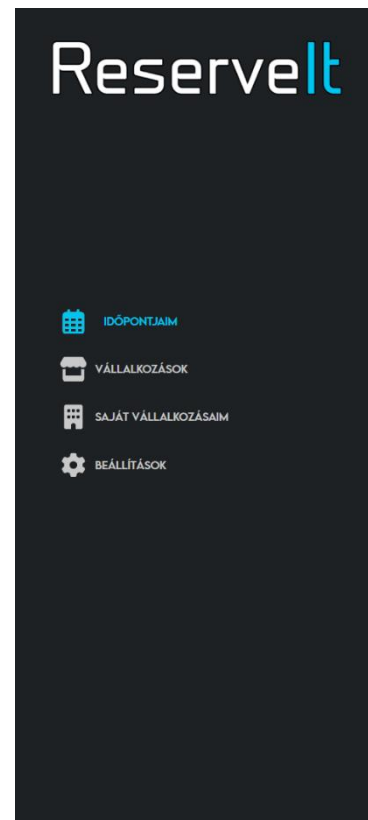
3. ábra Az elfelejtett jelszó felület



## 3.2. Kezelőfelület

### 3.2.1. Menü

A Menü képernyőfelbontástól függően jelenik meg a képernyőnk oldalán vagy a tetején, hogyha a készülékünk szélessége egy bizonyos értéknél alacsonyabb. Használata végtelenül egyszerű, a felhasználó rákattint a kívánt menüpontra és az oldal töltés nélkül átirányítja a kívánt menübe. Összesen négy menüpont található rajta ezek a következők: Időpontjaim, Vállalkozások, Saját vállalkozásaim és Beállítások. Ezekhez társítottunk ikonokat is. Ezt a menü komponenst felhasználtuk még a Vállalkozás kezelőnél is, viszont értelemszerűen más elnevezésekkel és azokhoz társított ikonokkal. A tervezés során figyeltünk arra, hogy a színek összhangban legyenek az oldal többi részével, így például, ha a kurzorunkat egy menüpontra húzzuk akkor az ahhoz tartozó logó és szöveg is a projekt logójában használt kék színre vált, illetve kissé jobbra húzódik. Természetesen ez egy animációval valósult meg, így nem tűnik úgy, mintha a szöveg csak úgy jobbra ugrana.



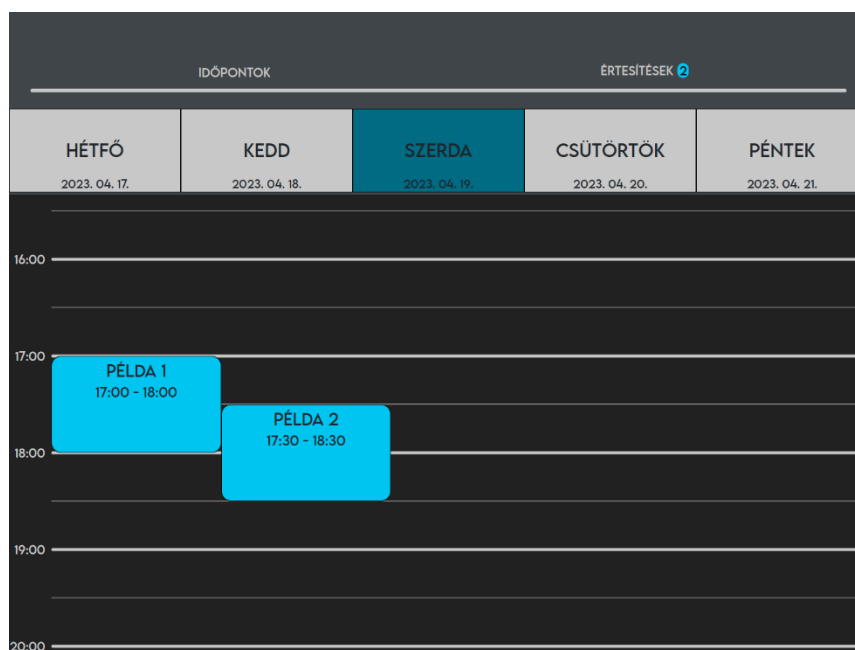
4. ábra A felhasználói menü

## 3.2.2. Időpontjaim

### 3.2.2.1. Időpontok

Ebben a menüben láthatjuk az időpontjainkat, amelyeket a vállalkozások adtak nekünk. Legfelül tudunk váltani a két almenü között, hogyha értesítést kaptunk, akkor a rendszer az Értesítések szöveg mellé helyezi az új értesítések számát. Alatta láthatók dátumok, amelyekre kattintva tudunk váltani a napok között. Ez alatt van az arra a napra vonatkozó értesítések helye, ez az órák szerint van bevonalmazva, illetve fél óránként egy vékonyabb vonal található. Az időpontok egyértelműen az ahhoz tartozó vonalnál jelennek meg, hogyha átfedés van két időpont között, akkor a rendszer egymás mellé fogja helyezni őket, ahogyan az ábrán látható. Az időpontban látható a neve és az is, hogy mettől – meddig tart. Ha rákattintunk, akkor egy panel fog megjelenni, amely tartalmazza az összes hozzá tartozó részletet. Ezt a panel komponenst használtuk fel a vállalkozások nézeténél is, így pontosan ugyanazokat az információkat látják, mint amit az időpontot kiosztó megadott. Ezek a mezők semmilyen módon nem módosíthatóak, kizárólag csak az időpontot kiosztó vállalkozás tagjai képesek rá.

5. ábra Időpont adatai

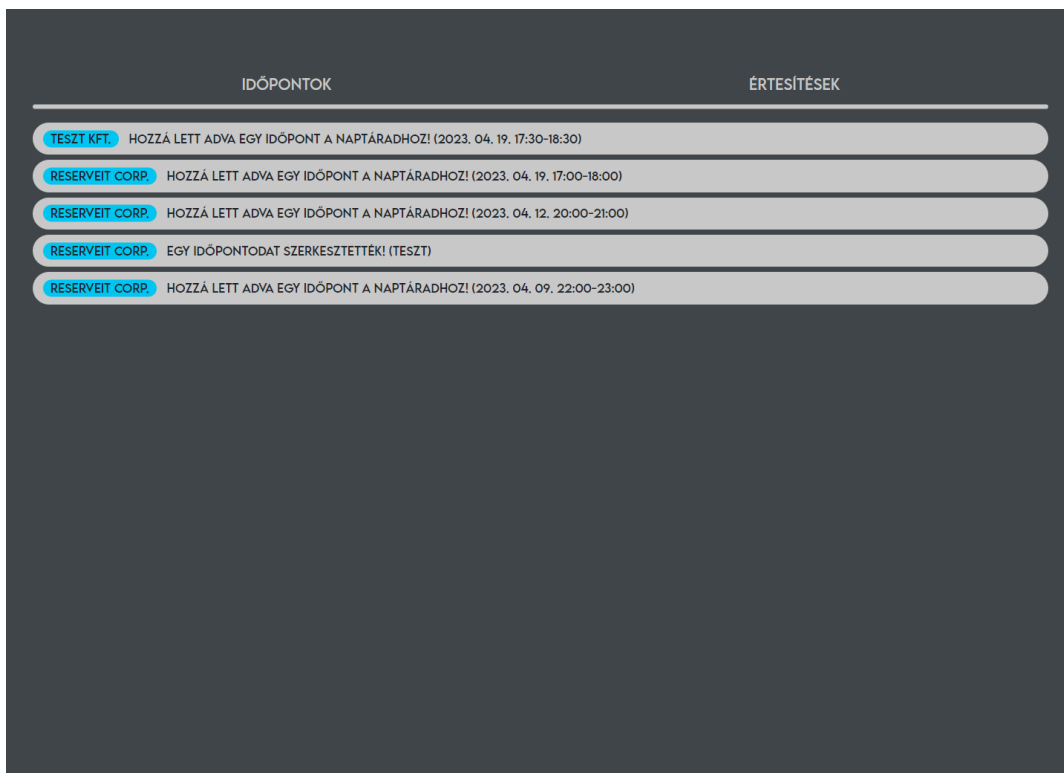


6. ábra A felhasználói naptár

### 3.2.2.2. Értesítések

Ebben a menüpontban láthatjuk az időpontjainkhoz tartozó értesítéseket. Az értesítésnél láthatjuk a vállalkozás nevét, mi történt és létrehozás esetén az időpontot, szerkesztés esetén a szerkesztett időpont nevét. Hogyha beléptünk az értesítések menüben, akkor az értesítések elolvasottnak számítanak, ezért nem jelzi már őket a felül, mint új. A sorrendjük az alapján van megadva, hogy mikor adták őket hozzá, szóval mindig a

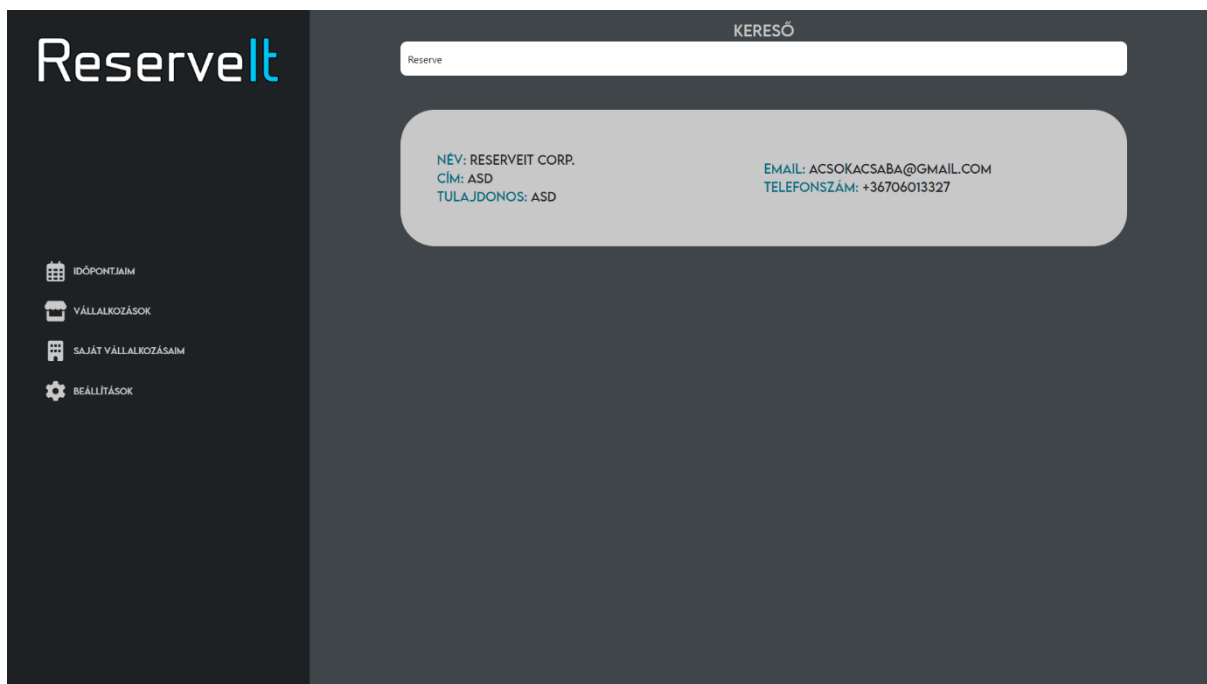
legfrissebb értesítés lesz legfelül.



7. ábra Az értesítések menü

### 3.2.3. Vállalkozások

Itt láthatjuk az összes vállalkozást, amely a rendszerben található és a hozzájuk tartozó elérhetőségeket, illetve a tulajdonosának a nevét. A kereső segítségével pedig egyértelműen keresni tudunk a vállalkozások között, amely a neveiket hasonlítja össze a beírt szavakkal.



8. ábra A vállalkozások menüpont

### 3.2.4. Saját vállalkozásaim

#### 3.2.4.1. Vállalkozásaim

Ebben a menüben láthatjuk a vállalkozásokat, amelyeknek tagjai vagyunk. Hogyha be szeretnénk lépni a vállalkozás

kezelőfelületébe, akkor

nincs más dolgunk,

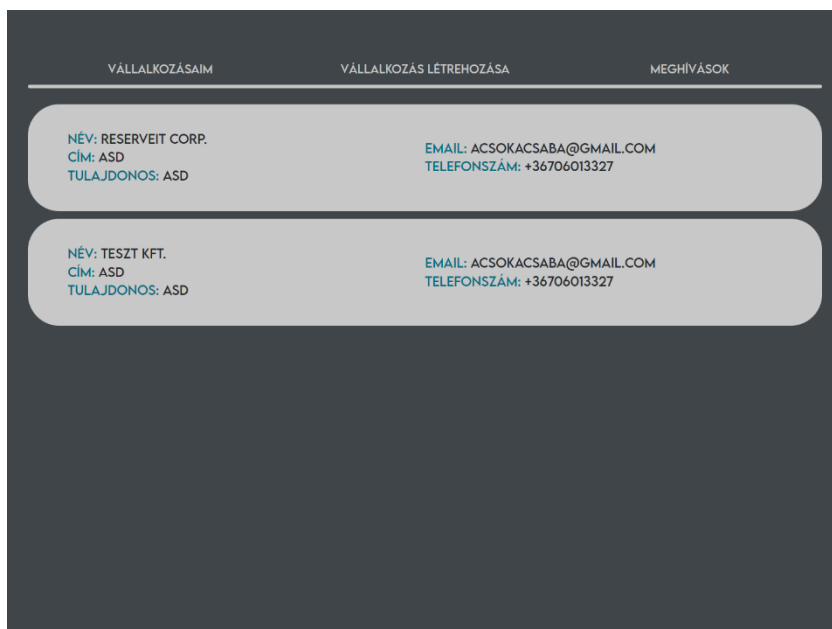
mint rákattintani a

kívánt szürke dobozra. Nincs arra külön

korlátozás, hogy egyszerre

hány vállalkozás tagjai

lehetünk. Abban az esetben, ha túl sok vállalkozás tagjai lennénk egyszerre, akkor csak



9. ábra A felhasználó vállalkozásai

simán le kell görgetnünk a kiválasztáshoz.

### 3.2.4.2. Vállalkozás létrehozása

Értelemszerűen itt tudunk saját vállalkozást létrehozni, amelyhez nincs más dolgunk, mint megadni annak nevét és elérhetőségeit, illetve a tulajdonos nevét és elérhetőségeit. A tervezés során szóba jött, hogy meg kelljen-e adni a

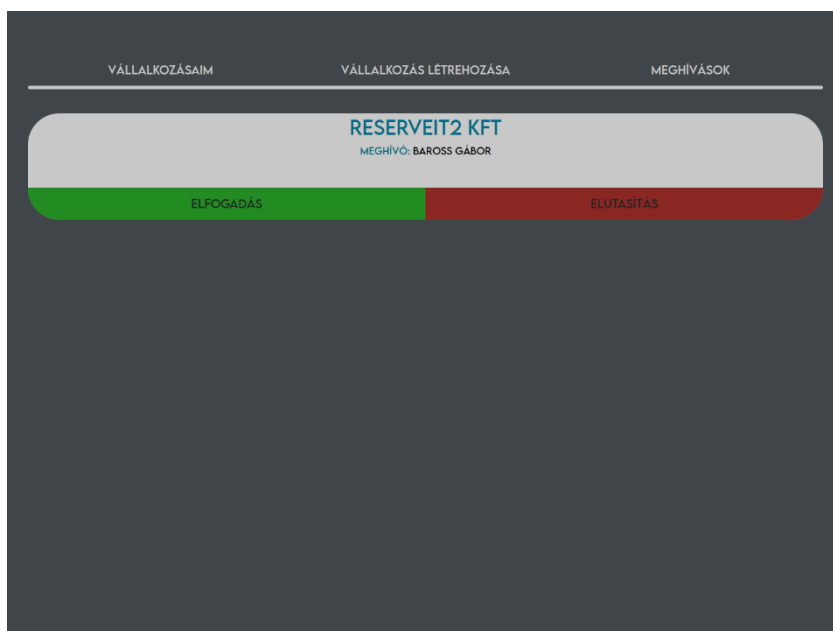
cégjegyzékszámot, viszont biztos ellenőrzési rendszer

10. ábra Vállalkozás létrehozása felület

hiányában ezt elvetettük. Ha valamilyen probléma lépne fel, akkor azt a rendszer az érintett mező alatt fogja kiírni vörössel. Ha sikeresen létrehoztuk a vállalkozást, akkor azt rögtön el lehet érni a Vállalkozásaim mező alatt.

### 3.2.4.3. Meghívások

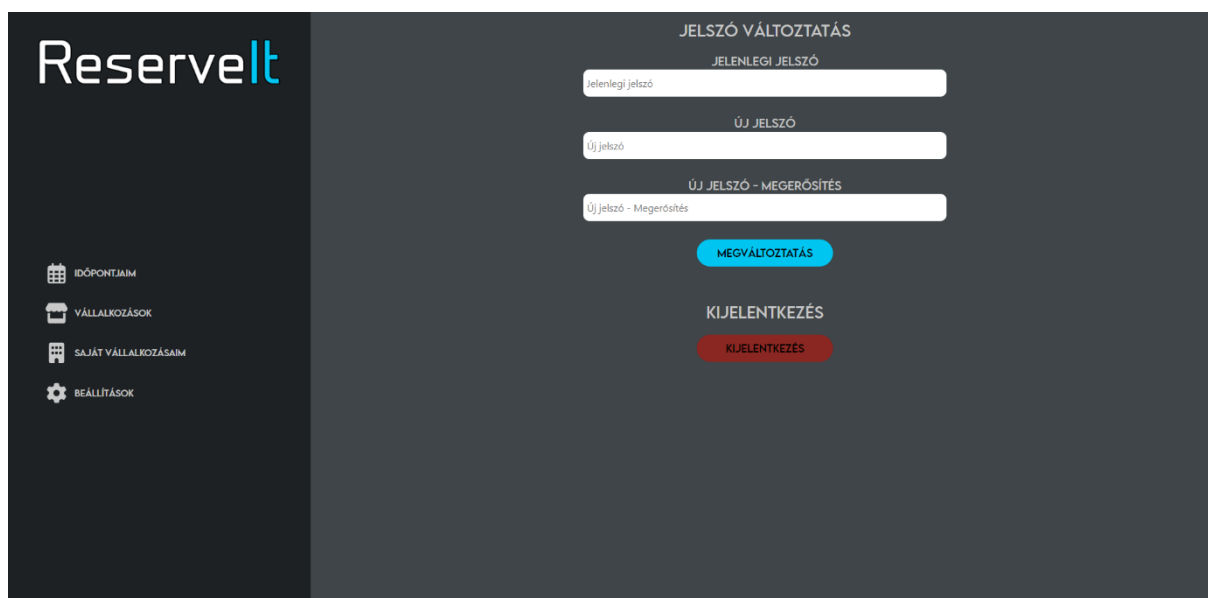
Itt tudjuk megtekinteni a meghívásokat a vállalkozásokhoz. Természetesen látjuk a vállalkozás nevét és aki meghívott minket, jelenleg ez minden esetben csak a vállalkozás létrehozója lehet. A kapott meghívásokat kezelni szimplán az elfogadás vagy elutasítás gombokkal tudjuk. A kapott meghívásokat a rendszer egymás alá fogja kilistázni.



11. ábra A felhasználó meghívásai vállalkozásokba

### 3.2.5. Beállítások

Ebben a menüpontban tudjuk megváltoztatni a jelszavunkat, illetve kijelentkezni. A



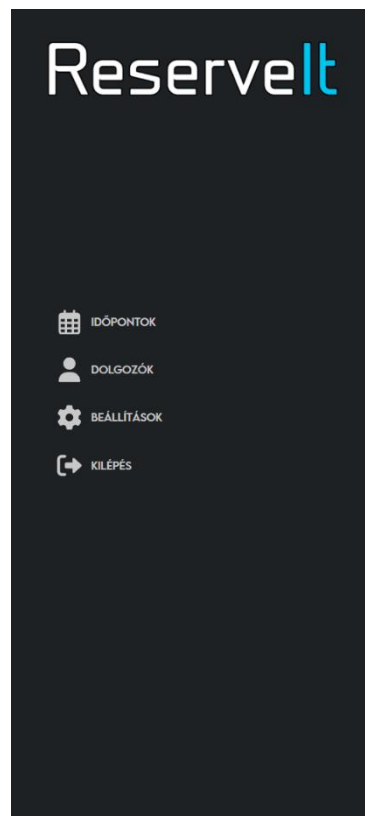
12. ábra A felhasználói beállítások

jelszó megváltoztatáshoz szükség van a jelenlegire, aztán pedig az új jelszót kétszer meg kell adni. A kijelentkezés pedig az ezzel a felirattal ellátott gomb megnyomásával tehető meg.

### 3.3. Vállalkozás kezelőfelület

#### 3.3.1. Menü

Hasonlóan a sima kezelőfelülethez itt is ugyanaz a menü komponens található meg, viszont más menüpontokkal. A Kijelentkezés menüpont annyiban különbözik, hogy nem visz egy másik menübe, hanem csak egyszerűen visszavisz minket a felhasználó kezelőfelületére.



13. ábra A vállalkozás  
kezelése menü



### 3.3.2. Időpontok

A sima kezelőfelületnél található időpontok fülhöz hasonlóan épül fel ez a menü is. Megtalálhatóak rajta a napok, illetve az ahhoz a naphoz tartozó órákra bontott naptár. Ha rákattintunk a naptár bármelyik pontjára, akkor meg fog jelenni az időpont hozzáadása fül, amelyben a szükséges adatok megadása után ki tudunk osztani időpontokat a vendég email címének tudatában. Az időpont hozzáadásnak olyan kikötései vannak, mint hogy nem lehet az időpont hamarabb, mint amikor ki lett adva, tartalmaznia kell nevet és kell hozzá rendelnie egy vendégnek. Ha megadtuk a vendék email címét, akkor a rendszer megkeresi a hozzá tartozó vezetés és keresztnévet, aztán pedig kitölti nekünk.

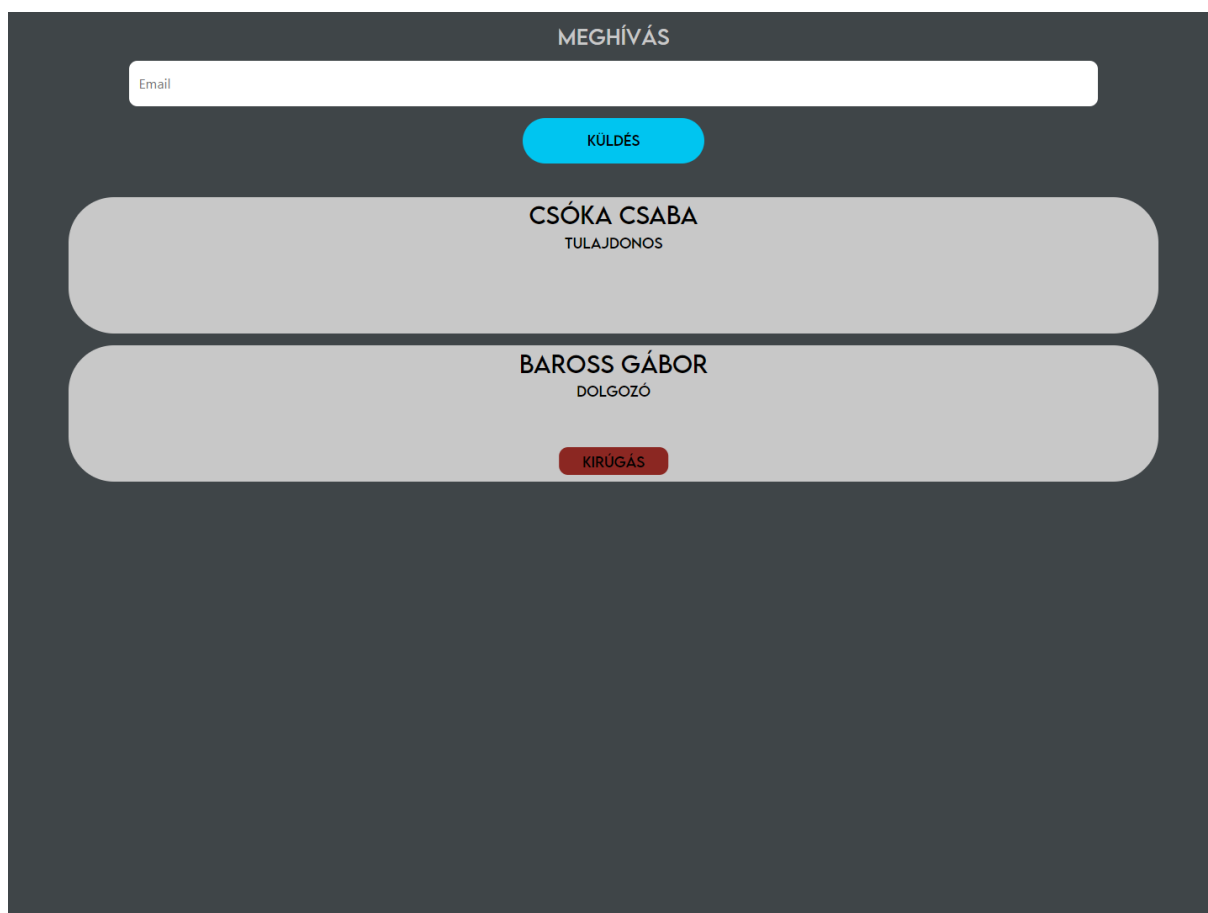
14. ábra Időpont létrehozása

HÉTFŐ 2023. 04. 17.	KEDD 2023. 04. 18.	SZERDA 2023. 04. 19.	CSÜTÖRTÖK 2023. 04. 20.	PÉNTEK 2023. 04. 21.
02:00	CSÓKA CSABA	BAROSS GÁBOR		
03:00				
04:00				
05:00				
06:00				
07:00				

15. ábra A vállalkozások naptára

### 3.3.3. Dolgozók

A vállalkozáshoz tartozó dolgozók itt vannak felsorolva. A vállalkozás tulajdonosának van joga meghívni, illetve kirúgni tagokat. A dolgozók nem látják a kirúgás



The screenshot displays a dark-themed interface titled "MEGHÍVÁS". At the top, there is a white input field labeled "Email" and a blue button labeled "KÜLDÉS". Below this, there are two user entries, each in a light gray rounded rectangle. The first entry is for "CSÓKA CSABA" with the role "TULAJDONOS" (Owner). The second entry is for "BAROSS GÁBOR" with the role "DOLGOZÓ" (Employee). Below the second entry, there is a red button labeled "KIRÚGÁS" (Dismiss).

16. ábra A Dolgozók menüpont

gombot és a meghívás mezőt.

### 3.3.4. Beállítások

Kizárólag csak a vállalkozás tulajdonosa látja a beállításokat. Ha dolgozó lép be ide, akkor egy szöveg fogja tájékoztatni róla, hogy nincs hozzáférése. Két beállítás érhető el, mind a kettőhöz szükséges a vállalkozás tulajdonosának jelszava. Ez a két beállítás pedig a vállalkozás nevének megváltoztatása, illetve annak törlése.

BEÁLLÍTÁSOK

JELSZAVAD

jelszavad

VÁLLALKOZÁS NEVÉNEK MEGVÁLTOZTATÁSA

ÚJ NÉV

Új név

MEGVÁLTOZTATÁS

VÁLLALKOZÁS TÖRLÉSE

TÖRLÉS

17. ábra A vállalkozások beállításai

## 4. Backend

A szerver NodeJS-t használ az adatbázissal való kommunikáció során. Minden adatlekérésre vagy szerkesztésre egy külön útvonalat hoztunk létre, természetesen, ha volt lehetőség, akkor egy útvonalat többször is felhasználtunk. Az átláthatóság miatt a routeoknak külön fájlokat hoztunk létre.

### 4.1. /addReservation

Itt lehet a vállalkozásoknak felvennie időpontokat a rendszerbe. Mindenekelőtt a felhasználónak be kell jelentkezve lennie az oldalon, máskülönben a szerver visszautasítja a kérést. Ellenőriz olyan dolgokat, mint hogy az időponthoz kötött ár szám-e, az időpont kezdete hamarabb van-e, mint a vége, az időpont később van-e, mint a jelenlegi idő, az időpontot hozzáadó a vállalkozás tagja-e, létezik-e a felhasználó, akinek felvették az időpontot, létezik-e a dolgozó, akihez felvették az időpontot és hogy a dolgozónak nincs-e már abban az időben egy időpontja. Ha minden rendben volt, akkor létrehozza az időpontot és egy értesítést a vendégnek, aki kapta az időpontot.

### 4.2. /changePassword

Ez az útvonal kicsit többoldalúbb, mivel egyszerre két jelszó változtatási módot is támogat. Ez a két mód pedig az elfelejtett jelszó és a jelszó megváltoztatása a jelenlegi jelszó tudatában. A két módot az különbözteti meg, hogy az elfelejtett jelszó esetében a kérés nem tartalmazza a jelenlegi jelszavát a felhasználónak, viszont az elfelejtett jelszó kulcsot igen. Természetesen ellenőrzi azt, hogy a jelszavak teljesítik-e a követelményeket, illetve, hogy nem-e egyeznek meg a felhasználó jelenleg használt jelszavával.

### 4.3. /checkNewPasswordKey

Itt kerül ellenőrzésre az új jelszó kód, amelyet emailben kapott, pontosabban, hogy létezik-e és hogy lejárt-e. Ha minden rendben van, akkor elküldi, hogy minden rendben volt. Ha mégse, akkor küldd egy hibaüzenetet az érintett problémát leírva.

### 4.4. /createOrganization

A vállalkozások létrehozása itt megy végbe. Természetesen be kell jelentkezve lenni a route használatához. Szükséges hozzá a vállalkozás neve, címe, email címe és a tulajdonos neve, címe és telefonszáma. Meg lehet még adni a vállalkozás telefonszámát, de nem kötelező. Ellenőrzi, hogy a vállalkozás neve foglalt-e és hogyha nem, akkor

létrehozza a rendszerben.

#### **4.5. /deleteOrganization**

Ahogy a név is sugallja, a vállalkozások törlése itt történik. Be kell jelentkezve lenni, illetve a törlést kérvényezőnek kell lennie a vállalkozás tulajdonosának és a kérésnek tartalmaznia kell a tulajdonos jelszavát biztonsági okokból. Ha minden rendben van, akkor a rendszer törli a vállalkozást, dolgozóit és a hozzá kapcsolódó foglalásokat.

#### **4.6. /editReservation**

Ha szerkesztjük valamelyik foglalást, akkor itt kerül majd feldolgozásra. Mivel a panel szinte teljesen ugyanaz, mint a létrehozásnál, ezért ugyanazok az adatok kellene, mintha létre akarnánk hozni egy foglalást, annyi különbséggel, hogy ehhez kell a szerkeszteni kívánt vállalkozás azonosítója. Ugyanazok az ellenőrzések lemennek, mint létrehozásnál és szerkesztésre is kerül az időpont, hogyha minden rendben van. Szintén küldd értesítést a felhasználónak, annyi különbséggel, hogy a szerkesztésről tájékoztatja a felhasználót és nem pedig a létrehozásról.

#### **4.7. /forgottenPassword**

Az elfelejtett jelszó kérés itt kerül feldolgozásra. A kérésnek tartalmaznia kell a felhasználó email címét. A rendszer ellenőrzi, hogy nem-e küldtek az elmúlt 10 percben kérést, különben elutasítja. Hogyha minden rendben volt, akkor a megadott email címre elküldésre kerül az elfelejtett jelszó megváltoztatásához link.

#### **4.8. /getAllOrganizations**

Ez a route adja vissza az összes rendszerben található vállalkozás adatait. Mindössze be kell lennünk jelentkezve az oldalra, hogy megkapjuk a listát.

#### **4.9. /getBusinessInvites**

Itt kérjük le a felhasználóhoz, amelybe be vagyunk jelentkezve a vállalkozás meghívókat.

#### **4.10. /getNotifications**

Itt kérjük le a felhasználóhoz, amelybe be vagyunk jelentkezve az időpontjainkhoz kapcsolódó értesítéseket, amelyek vagy létrehozások, vagy szerkesztések lehetnek.

#### 4.11. /getNotificationsCount

Ez az útvonal csak azért jött létre, hogy ne kelljen minden értesítés adatait lekérni, inkább csak azokat, amelyek újként vannak megjelölve és azoknak is csak az összegét.

#### 4.12. /getOrganizations

Itt kérjük le azokat a vállalkozásokat, amelyekhez a felhasználónk hozzá van adva, mint dolgozó vagy tulajdonos.

#### 4.13. /getReservations

Itt kérjük le azokat az időpontokat, amelyek a kérdésben szereplő vállalkozás azonosítójához tartoznak és az ugyanitt megadott napon vannak.

#### 4.14. /getUserFromEmail

Hogyha ennek a kérdésnek küldünk egy email címet és be vagyunk jelentkezve, akkor vissza fogja adni az ahhoz az email címhez tartozó vezetéket és keresztnévet, illetve az azonosítóját. Hogyha nem szerepel az email cím a rendszerünkben, akkor ezek helyett egy üres szöveget ad vissza.

#### 4.15. /getUserReservations

Itt kérjük le a felhasználónk foglalásait, amelyek a kérdésben szereplő napon vannak.

#### 4.16. /getWorkers

A kérdésnek mindössze a kívánt vállalkozás azonosítójára van szükség, illetve, hogy be legyünk jelentkezve és vissza fogja adni az ahhoz a vállalkozáshoz tartozó összes hozzáadott felhasználó nevét és hogy tulajdonosa-e a vállalkozásnak.

#### 4.17. /inviteWorker

Ha meghívunk valakit a vállalkozásunkba, akkor ezen kerül feldolgozásra. Szükséges elküldeni a vállalkozás azonosítóját, illetve a meghívni kívánt személy email címét. Mindenekelőtt ellenőrzi, hogy a kérést indító személy a vállalkozás tulajdonosa-e, aztán hogy van-e a rendszerünkben a megadott email címhez társítva felhasználó, végül pedig azt ellenőrzi, hogy a vállalkozás tagja-e a személy vagy meg van-e már hívva. Ha minden rendben volt, akkor elküldi a meghívót a megadott felhasználónak.

#### 4.18. /isOrganizationAuthorized

Itt azt ellenőrizzük, hogy a felhasználó a vállalkozás tagja-e. Ha nem, akkor egy erről tájékoztató választ küld a szerver. Hogyha igen, akkor azt küldi, illetve, hogy a vállalkozás tulajdonosa-e vagy sem.

#### 4.19. /loginUser

Ezen az útvonalon történik a felhasználók bejelentkeztetése. Itt csak az emailt és a jelszót kell küldeni. Ellenőrzi, hogy az email megtalálható-e a rendszerben, ha igen, akkor az a jelszó lett-e megadva, amely hozzá tartozik és még azt is, hogy olyan felhasználó-e, amely meg van-e erősítve. Hogyha minden rendben van, akkor frissíti az adatbázisban a felhasználó legutóbbi bejelentkezését jelző időt és feltölti a felhasználó böngészőjének sütijei közé a bejelentkezést jelző JWT token.

#### 4.20. /logout

Ezzel az útvonallal lehet kijelentkezni, ami abból áll, hogy a felhasználó böngészőjének süti tárolójából kitörli a JWT token, amely jelzi a rendszer felé, hogy be van jelentkezve.

#### 4.21. /recaptcha

Ez szolgál arra, hogy a bejelentkező felületen történt interakciókat ellenőrizze a reCAPTCHA rendszer. Szimplán egy token-t kell továbbítani erre az útvonalra és visszatérési értéknek visszaküldi, hogy ember végezte-e az interakciót.

#### 4.22. /registerUser

Ez az útvonal végzi el a felhasználók regisztrációját a rendszerbe. Mindenekelőtt ellenőrzi, hogy minden adat (Vezetéknév, Keresztnév, Email, Jelszó és megerősítő jelszó) meg van-e, amely a regisztrációhoz szükséges. Ellenőrzi még, hogy a két megadott jelszó egyezik-e, a jelszó, amelyet megadott teljesíti-e a követelményeket és hogy foglalt-e a megadott email cím. Ha minden rendben volt, akkor létrehozza a rendszerben a felhasználót és elküld egy megerősítő emailt.

#### 4.23. /removeReservation

A foglalások törlése ezen az útvonalon történik. Természetesen be kell jelentkezve lenni, illetve az érintett időpontot kiosztott vállalkozás tagjának kell lenni. A kérdésben el

kell küldeni a foglalás azonosítóját, majd a rendszer törölni fogja azt.

#### **4.24. /removeWorker**

Hogyha a vállalkozás tulajdonosa el akar távolítani valaki a vállalkozásból, akkor itt kerül feldolgozásra. Egyértelműen szükséges a vállalkozás azonosítója és az eltávolítani kívánt azonosítója, majd ellenőrzésre kerül, hogy a kérést küldő a vállalkozás tulajdonosa-e és hogy az eltávolítani kívánt személy a vállalkozás tagja-e. Ha minden rendben van, akkor a rendszer eltávolítja a vállalkozásból a célszemélyt.

#### **4.25. /respondToInvite**

Amikor kapunk egy meghívót egy vállalkozásba, akkor két választásunk van, elfogadni és elutasítani. Ha az elutasítást választjuk, akkor a rendszer ellenőrzi, hogy létezik-e a meghívó a vállalkozás azonosítója és a választ adó azonosítója alapján. Ha ezek rendben vannak, akkor a meghívó eltávolításra kerül a rendszerből. Hogyha elfogadjuk, akkor ugyanez történik, viszont azzal a különbséggel, hogy hozzáadja a meghívottat a vállalkozáshoz.

#### **4.26. /updateOrganizationName**

A vállalkozások neve itt kerül megváltoztatásra, amelyhez szükséges az új név és a vállalkozás tulajdonosának jelszava. Kizárólag csak a vállalkozás tulajdonosa fér hozzá. Ellenőrzésre kerül, hogy a vállalkozás neve foglalt-e, hogyha nem akkor a rendszer megváltoztatja azt.

#### **4.27. /verifyAccount**

Itt történik a felhasználók profiljának megerősítése. Ellenőrzi, hogy a megerősítő kód létezik-e a rendszerben, illetve hogyha létezik, akkor nem-e járt le. Természetesen, ha valamelyik feltétel nem teljesül, akkor hibát fog visszaküldeni.

#### **4.28. /verifyToken**

Ez az útvonal ellenőrzi, hogy a felhasználó böngészőjében lévő süti megfelelő adatokat tartalmaz-e. Pontosabban azt ellenőrzi, hogy a token azt az emailt tartalmazza, amelyhez a benne lévő felhasználó azonosítója van kötve.



## 5. Adatbázis

### 5.1. Reservelt\_Accounts

Legelőször egy olyan táblára volt szükség, amelyben tároljuk a felhasználók adatait. Ez a tábla a Reservelt\_Accounts nevet kapta. Ebben a táblában tároljuk a felhasználók azonosítóját (AccountID), email címét (Email), jelszavát titkosított formában (Password), keresztnévét (LastName), vezetéknévét (FirstName), telefonszámát (PhoneNumber), regisztrációjának idejét (RegistrationDate), legutóbbi bejelentkezésének idejét (lastLoginDate) és azt, hogy megerősítette-e az email címét (isEmailValidated).

### 5.2. Reservelt\_VerificationData

A következő tábla amire szükségünk volt az a Reservelt\_VerificationData, amely tárolja a felhasználók azonosítóját (AccountID), létrehozásának idejét (Time) és az ahhoz tartozó megerősítő kódot (VerificationID). Erre azért volt szükség, mivel amikor megkapja a megerősítő emailt, akkor a link amire kattintania kell tartalmazza a megerősítő kódot és a rendszer ennek a táblának a segítségével ellenőrzi a helyességét, illetve az adatbázisban tárolt idő alapján, hogy lejárt-e.

### 5.3. Reservelt\_ForgottenPasswordData

A következő volt a Reservelt\_ForgottenPasswordData, amely az előbbi táblához hasonlóan végez ellenőrzést, annyi különbséggel, hogy ez az elfelejtett jelszavakhoz tartozik. Három adatot tárol: a felhasználó azonosítóját (AccountID), az elfelejtett jelszóhoz szükséges kódot (VerificationID) és a kérés idejét (Time)

### 5.4. Reservelt\_Businesses

Miután befejeztük a bejelentkező felületet jött akkor a vállalkozások adatait tároló táblára volt szükségünk, ez lett a Reservelt\_Businesses. Ebben tároljuk a vállalkozások azonosítóját (BusinessID), neveit (Name), címüket (Address), üzleti email címét (BusinessEmail), üzleti telefonszámát (BusinessPhone), a tulajdonos nevét (OwnerName), a tulajdonos email címét (OwnerEmail), a tulajdonos telefonszámát (OwnerPhone), cégjegyzékszámot (RegistrationNumber), valamint az adószámot (TaxNumber). Az utóbbi kettő nem került felhasználásra semmilyen formában a megfelelő ellenőrsi módszer hiányában.

## 5.5. Reservelt\_BusinessInvites

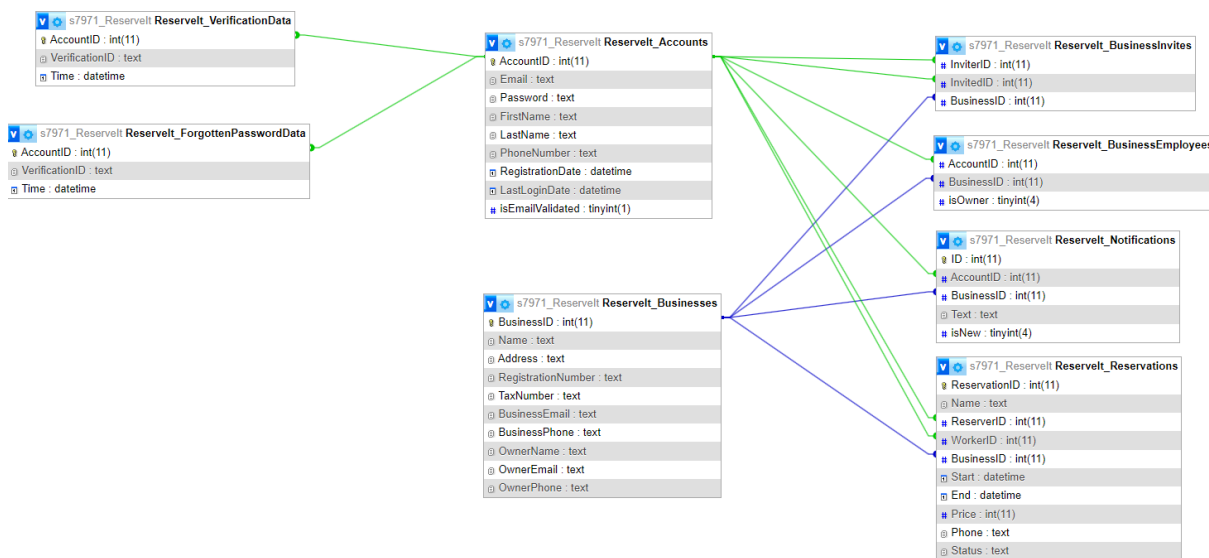
Ez a tábla tárolja a meghívókat a vállalkozásokhoz. Három dolgot tárol: a meghívó azonosítóját (InviterID), a meghívott azonosítóját (InvitedID) és a vállalkozás azonosítóját (BusinessID), amelybe meghívták. Ez a tábla nem tartalmaz elsődleges kulcsot.

## 5.6. Reservelt\_BusinessEmployees

Itt tároljuk a vállalkozások dolgozóit, illetve ez tartalmazza azt az értéket is, amely szerint a vállalkozás tulajdonosa-e a felhasználó. A következő adatokat tárolja: a dolgozó azonosítóját (AccountID), a vállalkozás azonosítóját (BusinessID) és hogy a vállalkozás tulajdonosa-e (isOwner).

## 5.7. Reservelt\_Reservations

Ebben tároljuk a foglalásokat, amelyeket a vállalkozások kiosztottak a vendégeknek. A következő adatokat tárolja: a foglalás azonosítóját (ReservationID), a foglalás nevét (Name), a vendéget, akié a foglalás (ReserverID), a dolgozót, akihez rendelték a foglalást (WorkerID), a foglalás kezdetét (Start), a foglalás végét (End), a foglalás árát (Price), a foglaláshoz rendelt telefonszámot (Phone), a foglalás állapotát (Status).



## **6. Tesztelés**

A backend tesztelés jest-el történik. Ha lefuttatunk egy tesztet az egész szerveren, akkor nem biztos, hogy elérjük a 100%-os lefedettséget mivel egy idő után az adatbázis nem enged több csatlakozást egyszerre, viszont eddig így is 95% fölött végződtek. Hogyha egyenként indítjuk el a teszt fájlokat, akkor láthatjuk, hogy semmilyen probléma nem áll fenn.

A weboldalt több felbontáson is teszteltük, de főként 1920x1080-as felbontáson lett fejlesztve. Az átnéзések során nem fedeztünk fel olyan problémát, amely kisebb felbontásokon akadályozná a weboldal használatát. A minimum támogatott szélességet 320 pixelnek vettük. Azt még fontos kiemelni, hogy a weboldal 980 pixel alatt vált úgymond mobil nézetbe. Ez főként a menü pozícióját változtatja meg, miszerint 980 pixel alatt a menü az oldal tetején található meg.

## **7. Továbbfejlesztési lehetőség**

### **7.1. Havi fizetés**

Ahogy fentebb is említettem, meg van benne a potenciál, hogy az oldal támogassa a fizetési lehetőségeket. Mi erre egy havi előfizetésre gondoltunk, amely inkább a vállalkozások tulajdonosait érintené. Konkrét előnyöket nem találtunk ki, de szóba jöttek olyanok, mint a vállalkozás kiemelése, amely a vállalkozások listájának a tetején kiemelve jeleníteni meg a vállalkozását az előfizetőnek.

### **7.2. Helyek**

Itt arra gondolunk, hogy a vállalkozásokhoz lehetne több helyet hozzáadni, szóval egy vállalkozás több helyen is lehetne egyszerre. A több helyhez lehetne más dolgozókat hozzáadni és külön kezelni őket a többiektől.

### **7.3. Térkép**

A térképen megjelenhetnének a vállalkozások pontos helyei, illetve így látnánk a hozzánk közeli vállalkozásokat. Lehetne feltüntetni a vállalkozások nyitvatartási idejét és egyéb információkat róla, mint a típusát.

### **7.4. Kuponok**

A weboldalon lehet megadni árat a foglaláshoz. Egy kupon rendszer segítségével lehetne kedvezményeket adni a vendégeknek, amelyeket az oldalon lehetne aktiválni. A vállalkozások természetesen maguk adnák ki a kuponokat, amelyek hozzájuk tartoznak.

### **7.5. Dolgozói jogrendszer**

Ez azt jelenti, hogy be lehetne állítani, hogy a dolgozóknak mihez van hozzáférése. Jelenleg a dolgozók tudnak felvenni időpontokat és törölni azokat, viszont nem tudnak meghívni vagy kirúgni más dolgozókat. Ez erre jelentene megoldást, a vállalkozás tulajdonosa tudná szabályozni a dolgozók jogosultságait igény szerint.

### **7.6. Statisztikák**

Lehetne egy külön menüpontot rakni a vállalkozások kezelőfelületéhez, amely az időpontokhoz kapcsolódó statisztikákat mutatna. Lehetne megjeleníteni a havi bevételt, kiosztott időpontokat vagy esetleg azt a dolgozót, aki a legtöbb órát teljesítette az időpontok alapján. Tervben volt hozzá adni a weboldalhoz, viszont időhiány miatt elvetésre

került.

## 8. Források

- <https://react.dev/>
- [https://en.wikipedia.org/wiki/React\\_\(software\)](https://en.wikipedia.org/wiki/React_(software))
- <https://nodejs.org/en>
- <https://en.wikipedia.org/wiki/Node.js>
- <https://expressjs.com/>
- <https://en.wikipedia.org/wiki/Express.js>
- <https://www.mysql.com/>
- <https://en.wikipedia.org/wiki/MySQL>
- <https://momentjs.com/>
- <https://en.wikipedia.org/wiki/CSS>
- <https://code.visualstudio.com/>
- [https://en.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://en.wikipedia.org/wiki/Visual_Studio_Code)
- <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- <https://en.wikipedia.org/wiki/JavaScript>
- <https://jestjs.io/>
- [https://en.wikipedia.org/wiki/Jest\\_\(framework\)](https://en.wikipedia.org/wiki/Jest_(framework))

## 9. Ábrajegyzék

1. ábra A regisztrációs felület .....	15
2. ábra A bejelentkező felület .....	15
3. ábra Az elfelejtett jelszó felület .....	16
4. ábra A felhasználói menü .....	17
5. ábra Időpont adatai .....	18
6. ábra A felhasználói naptár .....	19
7. ábra A felhasználó értesítései .....	<b>Hiba! A könyvjelző nem létezik.</b>
8. ábra A vállalkozások menüpont .....	21
9. ábra A felhasználó vállalkozásai .....	21
10. ábra Vállalkozás létrehozása felület .....	22
11. ábra A felhasználó meghívásai vállalkozásokba .....	23
12. ábra A felhasználói beállítások .....	23
13. ábra A vállalkozás kezelése menü .....	24
14. ábra Időpont létrehozása .....	25
15. ábra A vállalkozások naptára .....	25
16. ábra A Dolgozók menüpont .....	26
17. ábra A vállalkozások beállításai .....	27

## 10. Mellékletek

GitHub Repository - <https://github.com/a-csoka/ReserveIt>