



Eötvös Loránd Tudományegyetem
Informatikai Kar
Információs Rendszerek Tanszék

Car Service Booking

- Foglalási menedzser webes alkalmazás műhelyeknek

Szerző:

Szabó-Thalmeiner Bence
Programtervező informatikus BSc.

Témavezető:

Szalai-Gindl János-Márk
adjunktus, Ph.D.

Budapest, 2024

Tartalomjegyzék

| | | |
|-------|---|----|
| 1. | Bevezetés..... | 4 |
| 2. | Felhasználói dokumentáció | 6 |
| 2.1 | Áttekintés | 6 |
| 2.2 | Telepítési útmutató | 6 |
| 2.3 | Szükséges hardver és szoftver..... | 7 |
| 2.4 | Felhasználó típusok | 7 |
| 2.5 | Főoldal és Fejléc..... | 8 |
| 2.6 | Időpont foglalás..... | 10 |
| 2.6.1 | Helyszín választás | 10 |
| 2.6.2 | Export opciók | 11 |
| 2.6.3 | Foglaló Táblázat | 12 |
| 2.6.4 | Foglalás létrehozása | 13 |
| 2.6.5 | Foglalás törlése..... | 14 |
| 2.7 | Dolgozók | 16 |
| 2.7.1 | Főoldal..... | 16 |
| 2.7.2 | Dolgozó létrehozása és módosítása – Speciális jogosultságú funkció..... | 18 |
| 2.7.3 | Dolgozók törlése – Speciális jogosultságú funkció..... | 19 |
| 2.7 | Helyszínek..... | 20 |
| 2.7.1 | Főoldal..... | 20 |
| 2.7.2 | Helyszín részletei | 21 |
| 2.7.3 | Nagy térképes megtekintés..... | 21 |
| 2.7.4 | Helyszín menedzsment – Speciális jogosultságú funkció..... | 22 |
| 2.7.5 | Helyszín létrehozása – Speciális jogosultságú funkció..... | 24 |
| 2.7.6 | Helyszín módosítása – Speciális jogosultságú funkció..... | 25 |
| 2.7.7 | Helyszín törlése – Speciális jogosultságú funkció | 26 |

| | |
|---|----|
| 2.8 Elérhetőségek | 27 |
| 2.9 Fiókkezelő oldal – Speciális jogosultságú funkció | 28 |
| 2.10 Fiók adatok és menedzsment..... | 28 |
| 2.11 Fiók foglalási részletek..... | 31 |
| 2.12 Regisztráció és Bejelentkezés | 34 |
| 2.12.1 Regisztráció | 34 |
| 2.12.2 Bejelentkezés..... | 35 |
| 2.12.3 Elfelejtett jelszó..... | 36 |
| 2.12.4 Megerősítő email újra küldése | 38 |
| 2.13 Hiba- és egyéb jelzések | 39 |
| 2.13.1 Statikus hibák | 39 |
| 2.13.2 Toast értesítések | 40 |
| 3. Fejlesztői dokumentáció..... | 41 |
| 3.1 Áttekintés és Tervezés | 41 |
| 3.2 Felhasznált technológiák | 45 |
| 3.3 Rétegek..... | 45 |
| 3.3.1 Nézetek..... | 46 |
| 3.3.2 Modell | 50 |
| 3.3.3 Adat (Data) és Adatvándorlás (Migrations)– Kitérő | 52 |
| 3.3.4 Vezérlés (Controller) | 54 |
| 3.4 A főprogram..... | 57 |
| 3.5 Felhasználó kezelés | 57 |
| 3.5.1 Felépítés | 58 |
| 3.5.2 Szerepkörök..... | 58 |
| 3.5.3 Védelem..... | 59 |
| 3.6 Az adatbázis..... | 59 |
| 3.7 Tesztelés | 60 |

| | |
|---------------------------------------|----|
| 3.7.1 Szerkezet – Unit tesztelés..... | 60 |
| 3.7.2 Manuális tesztelés | 61 |
| 3.7.3 Érdekességek..... | 65 |
| 4. Összefoglalás..... | 66 |
| 5. Fejlesztési lehetőségek | 67 |
| 6. Forrásjegyzet | 68 |

1. Bevezetés

Az internet világában hatalmas átalakuláson esett át a kiszolgáló szektor: míg régen egy egyszerű időpontfoglalás rengeteg utánajárással, személyes találkozásokkal és tényleges papírok leadásával járt, manapság a világháló segítségével a legtöbb ügyünket egy gombnyomással el tudjuk rendezni, otthonunk kényelméből. Ennek ellenére sok üzlet még nem állt át a modernebb megoldásokra. Ez főleg azokra a szektorokra jellemző, amikben nem volt létszükséglet a rendszerek frissítése, így az üzemeltetők maradtak a jól bevált, de elavult módszereknél.

Munkám célja egy olyan rendszer kidolgozása, ami lényegesen leegyszerűsíti a foglalás menetét, első sorban járműszerelő műhelyek számára. Az alkalmazás egy ASP.NET alapú weboldal, ami egy adatbázis-kezelő segítségével számontartja a felhasználók foglalásait a rendszerben megtalálható műhelyekben. Célom a foglalás menetének és a rendszerben található foglalások, műhelyek és dolgozók menedzselésének egyszerűvé tétele.

A rendszer az MVC (modell, nézet, vezérlés) architektúrára alapul, melyben az alkalmazás felületi rétegét úgy nevezett "Razor Page"-ekre alapozzuk, amik ötvözik a C#, HTML, CSS és JavaScript nyelvek előnyeit egy a felhasználók számára barátságos, de emellett sokoldalú felület létrehozására. A modell réteg egy adatbázisból épül fel, amiben tároljuk a szükséges elemekre vonatkozó információkat, ebben bonyolítjuk le nem csak a műhelyek, munkások és foglalások kezelését, hanem a felhasználók adatainak tárolását is. A vezérlő réteg első sorban a c# nyelvre alapul, ebben végezzük el a bevett információk ellenőrzéseit, valamint a megjelenítéshez szükséges információk átadását.

A rendszerben szükségünk van a felhasználók kezelésre is. Alapvetően több típusú felhasználóra van szükségünk (adminisztrátor, egyszerű felhasználó, bejelentkezés előtt vendég), az adott típus szerint a weboldal funkcionalitása változhat (egyszerű felhasználóknak lehetőségük van új foglalások létrehozására, valamint a honlapon tárolt adatok megtekintésére, saját, már meglévő foglalásaik kezelésére, míg az adminisztrátorok új műhelyeket, munkásokat vehetnek fel, megváltoztathatják már meglévő elemek adatait, a vendégek is megtekinthetnek információkat, de nem foglalhatnak). Fontos szempont, hogy a felhasználók ne tudjanak egymás foglalásaiban kárt tenni, ne tudják más felhasználók foglalásait törölni.

A weboldalt megtámogatjuk néhány további kiegészítő lehetőséggel, ilyen a felhasználó foglalásai alapján való pdf generálása, a foglaló táblázat alapján egy Excel táblázat generálása, valamint beépülő modulok használata (ilyen a Google Maps integrálása, a Toast értesítések használata, vagy további modulok integrálása a felhasználói élmény javításának érdekében).

A tervezés fő célja egy olyan felület kialakítása volt, aminek a használata könnyen megtanulható, de nem kompromittálja a funkcionalitást. Célom az, hogy a foglалás minél több lépését egyszerű, könnyen érthető darabokra osszuk fel, hogy a felhasználó soha ne érezze magát elveszve, de emellett a háttérbe egy teljes értékű foglалási rendszert helyezünk, ami alapján minden lényeges információ a rendelkezésünkre áll, így mind az egyszerű felhasználóknak, mind az adminisztrátoroknak egy hasznos eszközt tudunk nyújtani, ami nagyban megkönnyíti a kommunikációt az ügyfelek és a szolgáltatók között.

2. Felhasználói dokumentáció

2.1 Áttekintés

A Car Service Manager applikáció, mint ahogy a bevezetőben is említettük, fő célja a műhelyekben folyó munkálatokhoz való időpontfoglalás leegyszerűsítése, valamint ezek egyszerű menedzselése. Ennek érdekében a rendszer több fő funkcióval is szolgál az átlag felhasználónak:

- Időpontfoglalás, időpont menedzsment.
- Helyszínek megtekintése.
- Dolgozók megtekintése.

Általában ezt a weboldalt bárki használhatja, akinek valamilyen gondja akadt járművével, és javításokat szeretne végezni rajta, így nem feltételezünk semmilyen magasabb számítógépes ismeretet a legalapvetőbb kompetenciák kivételével (böngésző használata, felhasználói fiókok fogalma).

Ezek mellett lehetőséget nyújtunk a műhelyek üzemeltetőinek is az adatok mélyebb módosítására, hogy helyes és friss adatokat találhassunk a weboldalon. A magasabb szintű felhasználóktól sem várunk el sokkal komolyabb kompetenciákat, de azért feltételezzük, hogy egyszerű adatok megadására képesek, valamint alapszintű angol nyelvi ismeretek birtokában vannak. Ezeknek a speciális felhasználóknak egyedi szerepköröket adunk (lásd 2.4 – Felhasználó típusok), a következő funkciók valamely kombinációját kapják meg:

- Helyszínek szerkesztése, menedzselése.
- Dolgozók adatainak szerkesztése, menedzselése.
- Felhasználói fiókok szerepköreinek megváltoztatása.

Az applikáció angol nyelven íródott, mivel akár globális alkalmazásra is lehetővé szeretnénk tenni.

2.2 Telepítési útmutató

A telepítés előtt mindenképp fel kell telepítenünk a Microsoft SQL server szolgáltatást az adott rendszerre. Ez elindíthatóvá válik az SQL Server 2022 Configuration Manager program segítségével [2].

Az alkalmazást a projekt futtatható állományának letöltése után az „AutonetProjectMVCAASP” futtatható állományra klikkelve tudjuk elindítani, feltéve, hogy előtte felépítettük az adatbázist (erről az alternatív indítás első futtatásánál bővebben beszélünk). Ezután a “localhost:5000” webcím beírásával elérhetjük az oldalt az általunk preferált böngészőn keresztül (érdemes Opera vagy Google Chrome böngészőkön keresztül használni, a fejlesztés ezeken a böngészőkön történt).

Alternatívan, ha esetleg mélyebben bele szeretnénk látni a program működésébe, valamint el szeretnénk érni a tesztelési projektet is, az alkalmazás futtatását a Visual Studio 2022 Fejlesztői környezet és a Microsoft SQL szerver szolgáltatás segítségével történik. A Visual Studio egyszerűen telepíthető a saját weboldala alapján [1], míg az SQL szerver elindítható az SQL Server 2022 Configuration Manager segítségével [2]. Opcionálisan letölthetjük a Microsoft SQL Server Manager Studio (SSMS) alkalmazást [3] az adatbázis direkt módú kezeléséhez.

A futtatás szintén nem komplikált. Miután elindítottuk az SQL szervert és szinkronizáltuk a szükséges dependenciákat (NuGet csomagok, ezeket a Visual Studio automatikusan letölti a projekt fájlok alapján), ki kell adnunk az Update-Database parancsot a Package Manager Console-ba (elérés: Tools > NuGet Package Manager > Package Manager Console) az adatbázis felépítéséhez. Ezt a lépést csak az első indításnál kötelező elvégezzük. Ha minden készen áll, akkor a futtatás gombra kattintva el is indíthatjuk az alkalmazást az alapértelmezett böngésző segítségével.

2.3 Szükséges hardver és szoftver

A Car Service Manager alkalmazás nem igényel különösebben erős hardver-t az alkalmazás megfelelő futtatásához. Ha az adott eszköz képes futtatni a fent említett alkalmazásokat és technológiákat (Visual Studio, SQL szerver), akkor az alkalmazás is feltételezhetően megfelelően fog futni.

Szoftver szempontjából a fent említett programok mellett szükségünk lesz még egy böngésző ablakra.

2.4 Felhasználó típusok

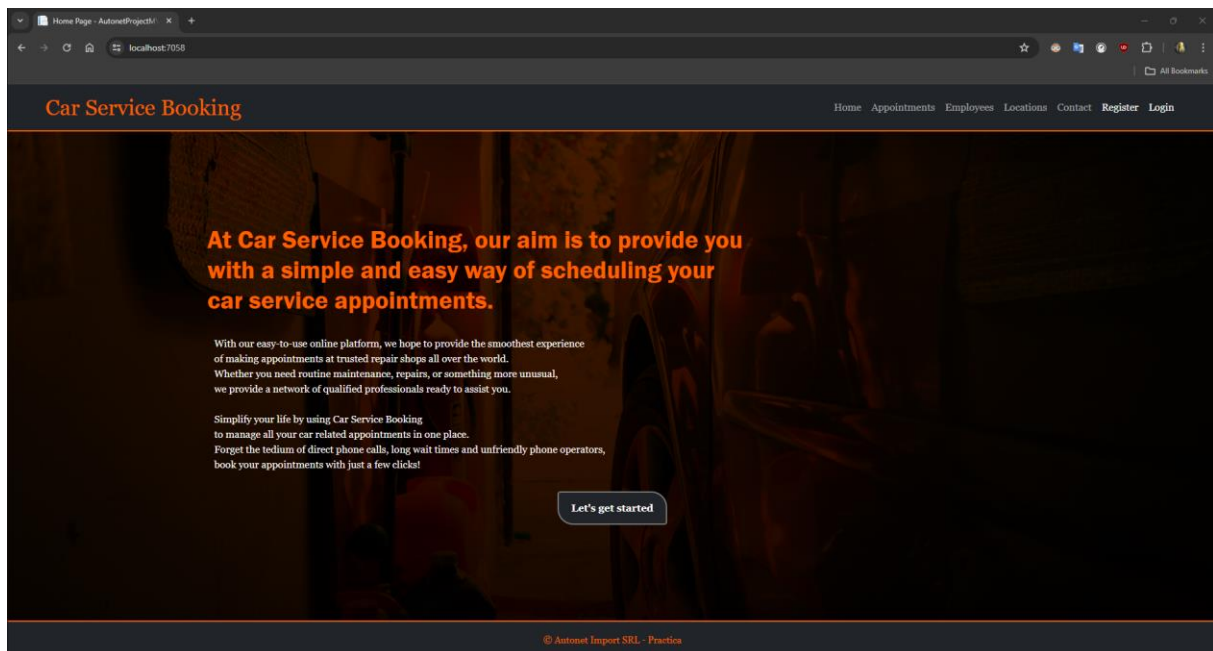
A Car Service Manager webalkalmazás rendelkezik fiókmenedzselési eszközökkel, így a felhasználó képes fiókot létrehozni. Minden fiók rendelkezik bizonyos jogosultságokkal, valamint a nem bejelentkezett személyek is afféle vendégként vannak számontartva:

- Vendég – Ebben a szerepkörben vagyunk bejelentkezés előtt. A vendég felhasználó megtekintheti a weboldal nagy részét, de nem csinálhat semmilyen olyan tevékenységet, amivel változna a kiírt információk állapota (például nem tud időpontot foglalni).
- Felhasználó – Ez az alapszintű felhasználó jogosultság, a továbbiakban nagyrészt ezt fogjuk bemutatni. A felhasználó képes új foglalás létrehozására, saját foglalásainak törlésére és módosítására, valamint saját felhasználói oldalainak megtekintésére. Ezeket később tüzetesen bemutatjuk.
- Főnök – Ezzel a jogosultsággal a felhasználó jogai mellett képesek vagyunk a dolgozók menedzselésére (létrehozás, módosítás, törlés)
- Tervező - Ezzel a jogosultsággal a felhasználó jogai mellett képesek vagyunk a helyszínek menedzselésére (létrehozás, módosítás, törlés)
- Admin – Ez a teljesen átfogó jogosultság. Az összes fent említett lehetőség mellett le tudja kérni az adatbázis teljes tartalmát, valamint egyedi jelzést is kap, mikor bejelentkezik (lásd 1. ábra).

Hello admin@admin.com! Admin

1. ábra: Rendszergazda köszöntő üzenet

2.5 Főoldal és Fejléc



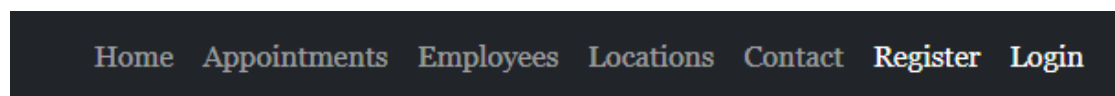
2. ábra: A főoldal kinézete bejelentkezés előtt

A weboldal elindításával a főoldalra kerülünk (lásd 2. ábra). Ezen az oldalon egy rövid leírással összegezzük az alkalmazás szerepét, valamint az oldal végén levő gombra kattintva elkezdhetjük a foglalást.

Ezek mellett megjelenik a fejléc is, ami az egyszerű navigációt teszi lehetővé. A fejléc a következő elemekből áll:

Car Service Booking

3. ábra: Fejléc cím

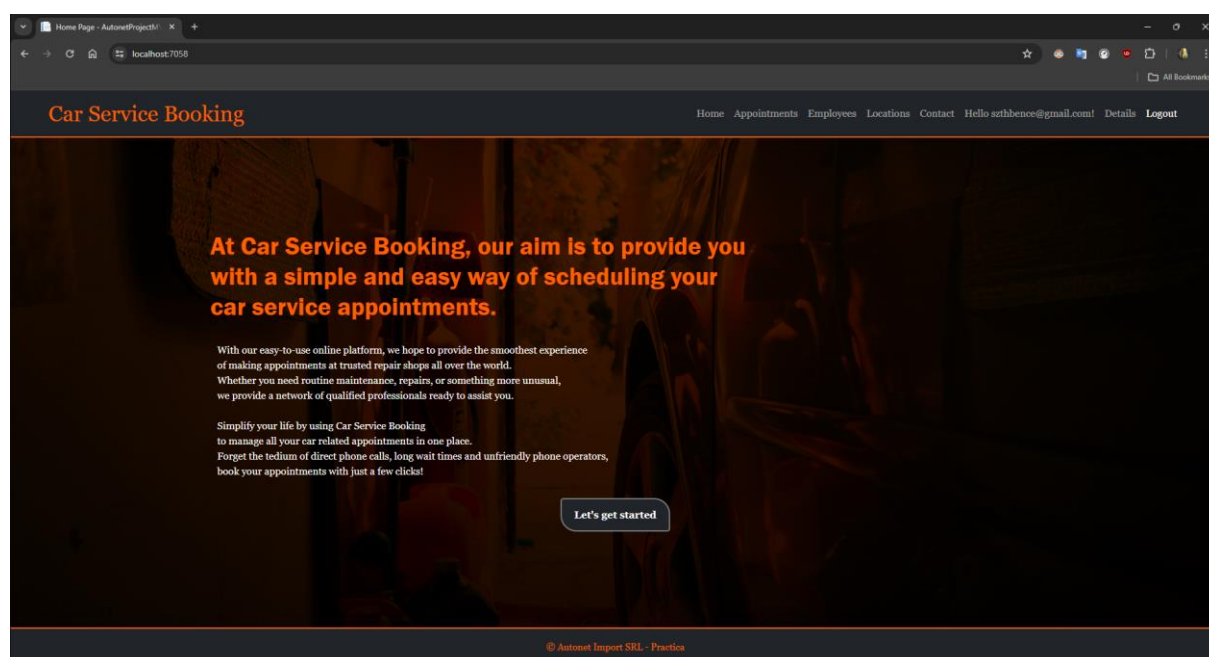


4. ábra: Fejléc gombok kinézete belépés előtt

- Cím és Home gomb – Ezzel a főoldalra kerülünk.
- Appointments – Ezzel az időpontfoglaló.
- Employees – A dolgozókat kiíró felület elérése.
- Locations – A helyszín listázó felület elérése.
- Contact – A műhelyekhez tartozó rövid információkat listázó felület elérése.
- Register – Regisztrációs felület.
- Login – Bejelentkezési felület.

A fentieket megfigyelhetjük a 3. ábrán és a 4. ábrán.

Meg kell jegyezzük, hogy bejelentkezés után változik a fejléc kinézete (lásd 5. ábra):



5. ábra: Főoldal kinézete bejelentkezés után

Itt két új gombot találunk (lásd 6. ábra):

Home Appointments Employees Locations Contact Hello szthbence@gmail.com! Details Logout

6. ábra: Fejléc gombok bejelentkezés után

- Felhasználó köszöntő –Fiókmenedzsment részleg
- Details – Saját foglалásokat listázó részleg
- Logout - Kijelentkezés

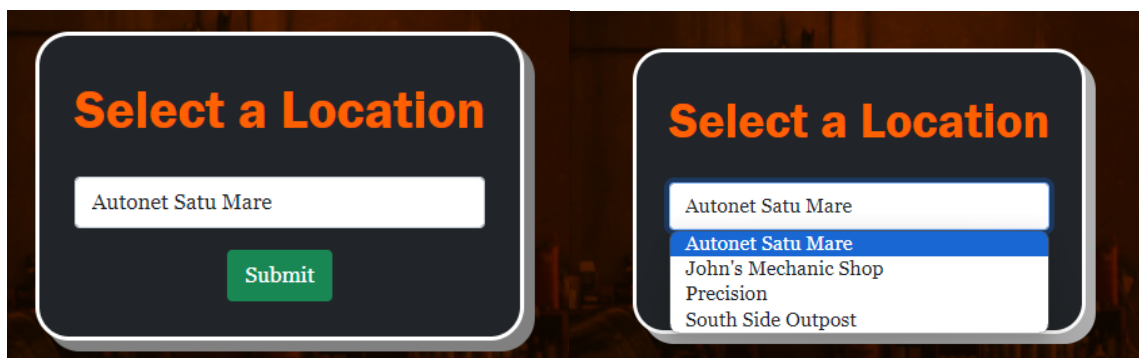
A fejléc mindig a képernyőnk felső részén lesz, még az után is, hogy legörgettünk, ezzel biztosítjuk, hogy a navigálás minél könnyebb legyen.

2.6 Időpont foglalás

Az alkalmazás első sorban azért készült, hogy egyszerűen és gyorsan hajthassunk végre időpontfoglalást egy általunk választott műhelyben, emiatt nagyon fontos volt, hogy ez a funkció a lehető legfejlettebb eleme legyen a programunknak. A foglalás során minden lépésben külön oldalra kerülünk, ezzel könnyen érthetővé tesszük a foglalás menetét.

2.6.1 Helyszín választás

Az “Appointments” gomb megnyomása után a következő felület fogad minket (lásd 7. ábra):

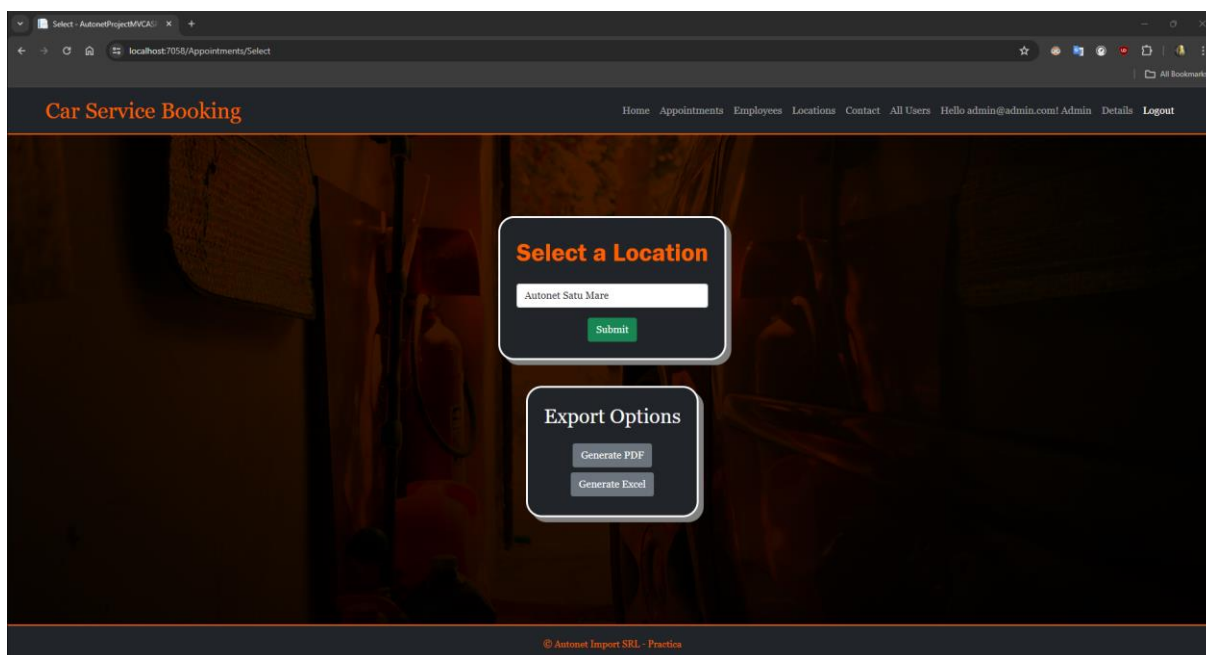


7. ábra: Select oldal alapállapotban és Select oldal a legördülő menüvel

A kártyán egy lenyíló menüt találunk, amivel kiválaszthatjuk a kívánt helyszínt, majd a zöld „Submit” gomb megnyomásával a foglaló felületre kerülünk.

2.6.2 Export opciók

Adminisztrátor módban az oldal kissé átalakul (lásd 8. ábra):



8. ábra: A helyszín választó oldal adminisztrátor módban

Megjelennek az adatkiviteli opciók, amikkel az adatbázis teljes tartalmát el tudjuk menteni egy Excel (lásd 9. ábra) vagy egy pdf (lásd 10. ábra) fájlba.

| | A | B | C | D | E | F | G | |
|---|----|-----------|----------|-----------|-----------|----------|-------|--|
| 1 | Id | Name | Time | Location | UserId | Employee | JobId | |
| 2 | 39 | Problem c | 45391.63 | Autonet S | admin@ | 5 | 46 | |
| 3 | 40 | Problem t | 45391.67 | Autonet S | admin@ | 5 | 47 | |
| 4 | 41 | Problem t | 45392.5 | John's Me | admin@ | 5 | 48 | |
| 5 | 42 | Problem f | 45391.58 | John's Me | szthbence | 5 | 49 | |
| 6 | 43 | Problem S | 45393.58 | Autonet S | admin@ | 5 | 50 | |
| 7 | 44 | Problem F | 45393.58 | Precision | admin@ | -1 | 51 | |

9. ábra: Az oldalon készített összes foglalás táblázata

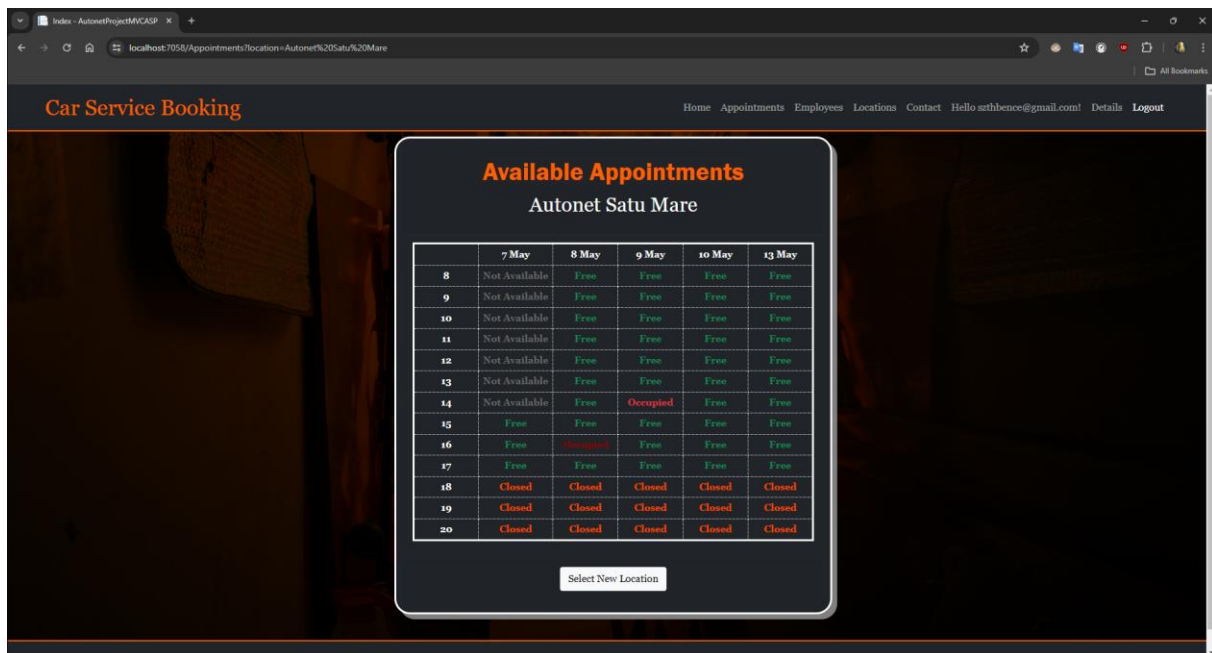
10. ábra:
Az oldalon
készített
összes
foglalás
pdfben

All appointments in the database

| Id | Appointment Name | Time of Appointment | Location | User | Employee |
|----|--------------------|---------------------|----------------------|---------------------|-------------|
| 39 | Problem one | 09/04/2024 15:00:00 | Autonet Satu Mare | admin@admin.com | Doe |
| 40 | Problem two | 09/04/2024 16:00:00 | Autonet Satu Mare | admin@admin.com | Doe |
| 41 | Problem three | 10/04/2024 12:00:00 | John's Mechanic Shop | admin@admin.com | Doe |
| 42 | Problem four | 09/04/2024 14:00:00 | John's Mechanic Shop | szthbence@gmail.com | Doe |
| 43 | Problem Seventy | 11/04/2024 14:00:00 | Autonet Satu Mare | admin@admin.com | Doe |
| 44 | Problem Forty-four | 11/04/2024 14:00:00 | Precision | admin@admin.com | No Employee |

2.6.3 Fogláló Táblázat

A helyszín kiválasztása után a következő oldal fogad minket (lásd 11. ábra):



11. ábra: Fogláló felület

Az oldal tetején megtaláljuk a címet, valamint a jelen helyszín nevét.

Ezen oldal fő eleme a táblázat, amiben az elkövetkező öt munkanapra tudunk egy órás időpontokat foglalni. A táblázat 5 fő elemből áll:

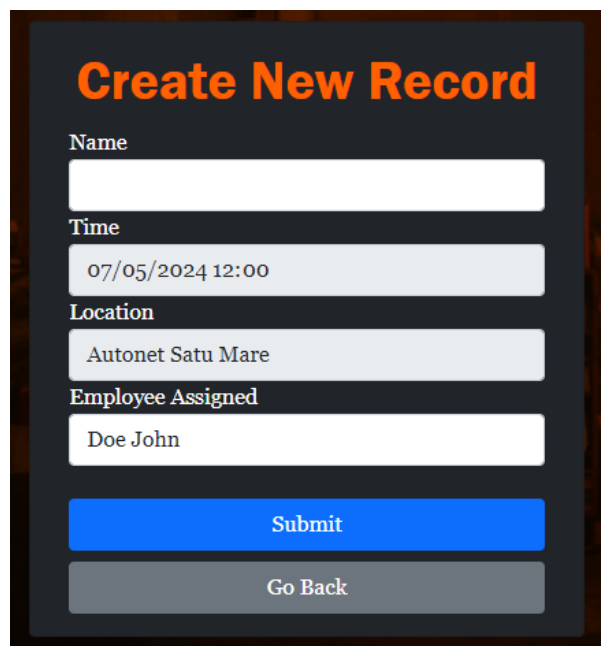
- Free – Ez az időpont szabad, klikkeléssel foglalhatunk, feltéve, hogy be vagyunk jelentkezve.
- Occupied – Ez 2 jelentéssel bírhat:
 - Világos – Ez a mi foglalásunk, klikkeléssel elindíthatjuk a lemondást.
 - Sötét – Ez más felhasználó foglalása.
- Closed – A műhely nem fogad ebben az időpontban.
- Not Available – Az adott időpont már a múltban van, túl későn próbáljuk lefoglalni.

2.6.4 Foglalás létrehozása

Egy lehetséges szabad elem megnyomása után a következő felülettel találkozunk (lásd 12. ábra):

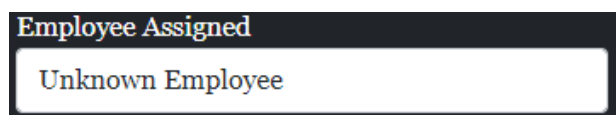
A következő elemeket állíthatjuk:

- Name – Foglalás neve (érdemes értelmes, leíró nevet adni, hogy sejtjük, mi a baj)
- Time – A táblázat alapján előre beillesztve
- Location – Ugyancsak a táblázat alapján
- Employee Assigned – Ki dolgozzon rajta?



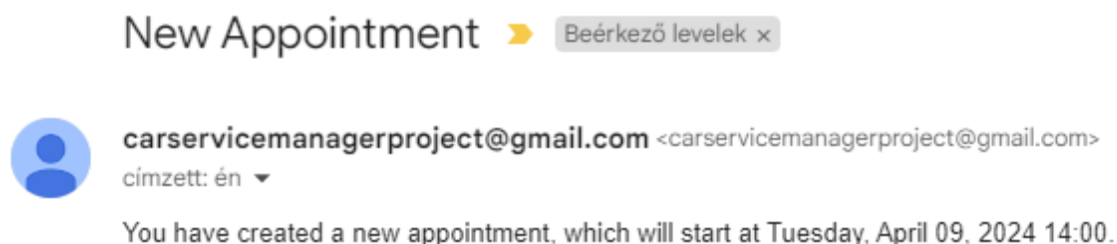
12. ábra: Foglalás készítő felület

Ha a műhely nem adott meg dolgozókat, erről is jelzést kapunk (lásd 13. ábra).



13. ábra: Dolgozó választó, ha nincs dolgozó a műhelyhez kapcsolva

Sikeres foglalás esetén Toast értesítés tudatja velünk, hogy minden rendben, valamint üzenetet kapunk a fiókhoz tartozó email-címre (lásd 14. ábra).

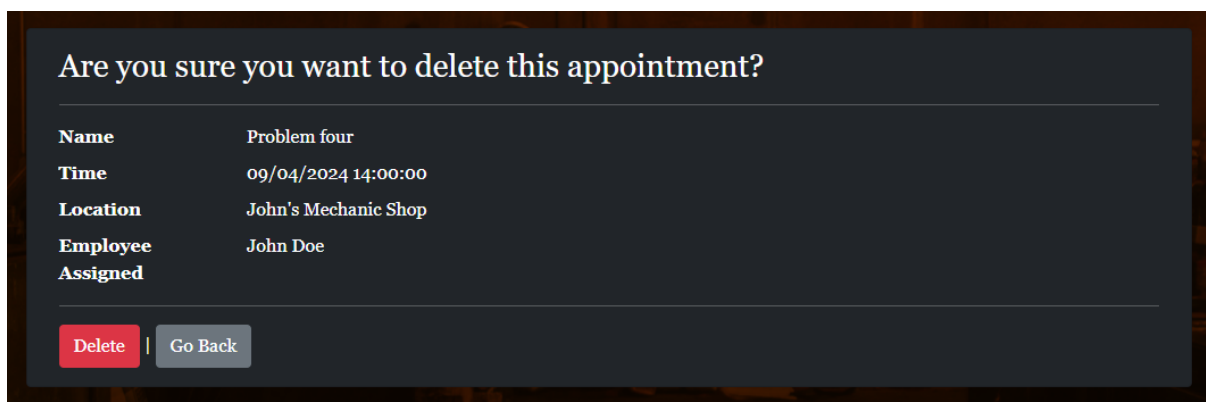


14. ábra: Foglalás email

Lehetőségünk van menteni, vagy visszamenni az előző oldalra. Az oldal emellett egy nappal az időpont előtt küldeni fog egy emlékeztető üzenetet az adott email-címre.

2.6.5 Foglalás törlése

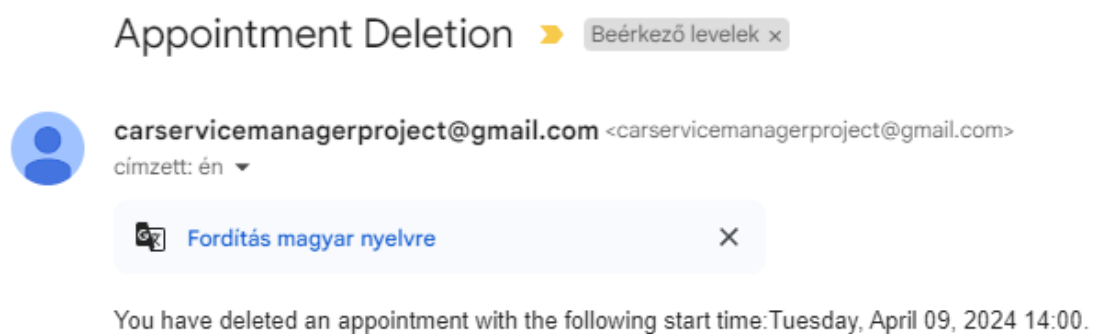
Ha saját foglalásunkra klikkelünk, a következő oldalt kapjuk (lásd 15. ábra):



15. ábra: Foglalás törlési felület

Itt megkapjuk a foglalás adatait, valamint a visszatérés és törlés gombokat, így lehetőségünk van félbeszakítani a műveletet. Itt meg kell jegyeznünk, hogy az Adminisztrátor joggal rendelkező fiókok bárkinek a foglalását kitörölhetik.

Az időpont törlése után figyelmeztető emailt kapunk a tevékenységről (lásd 16. ábra):

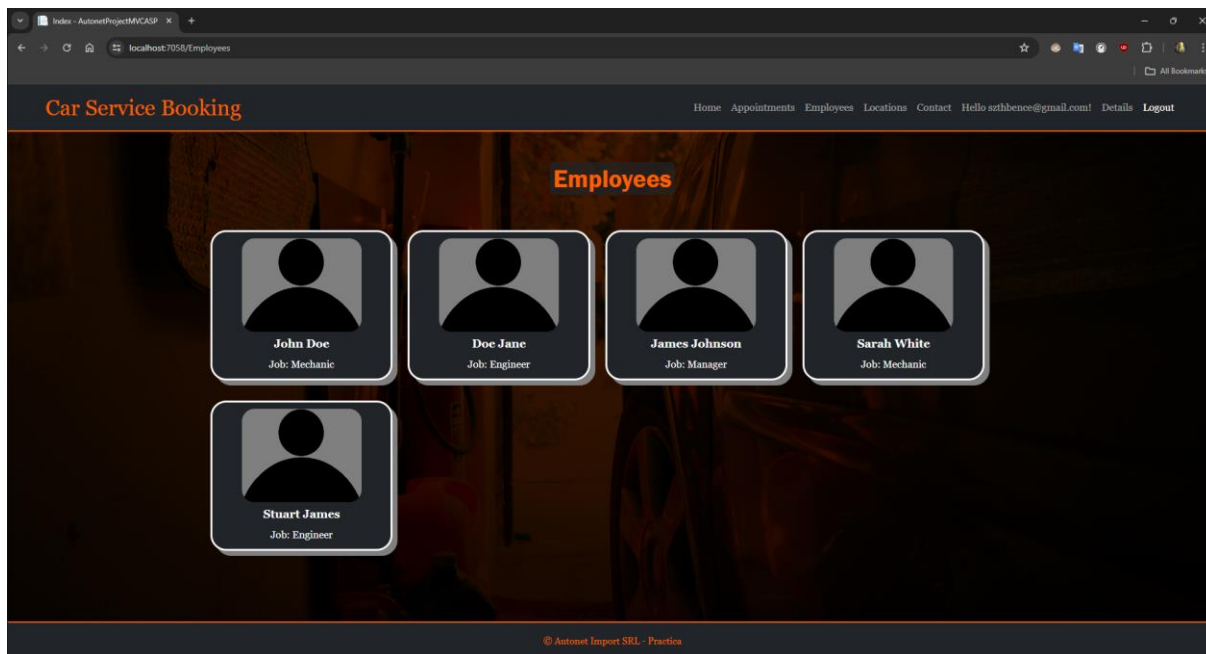


16. ábra: Foglálás törlés email

2.7 Dolgozók

2.7.1 Főoldal

Az „Employees” menüpontra kattintva a következő oldal fogad minket (lásd 18. ábra):

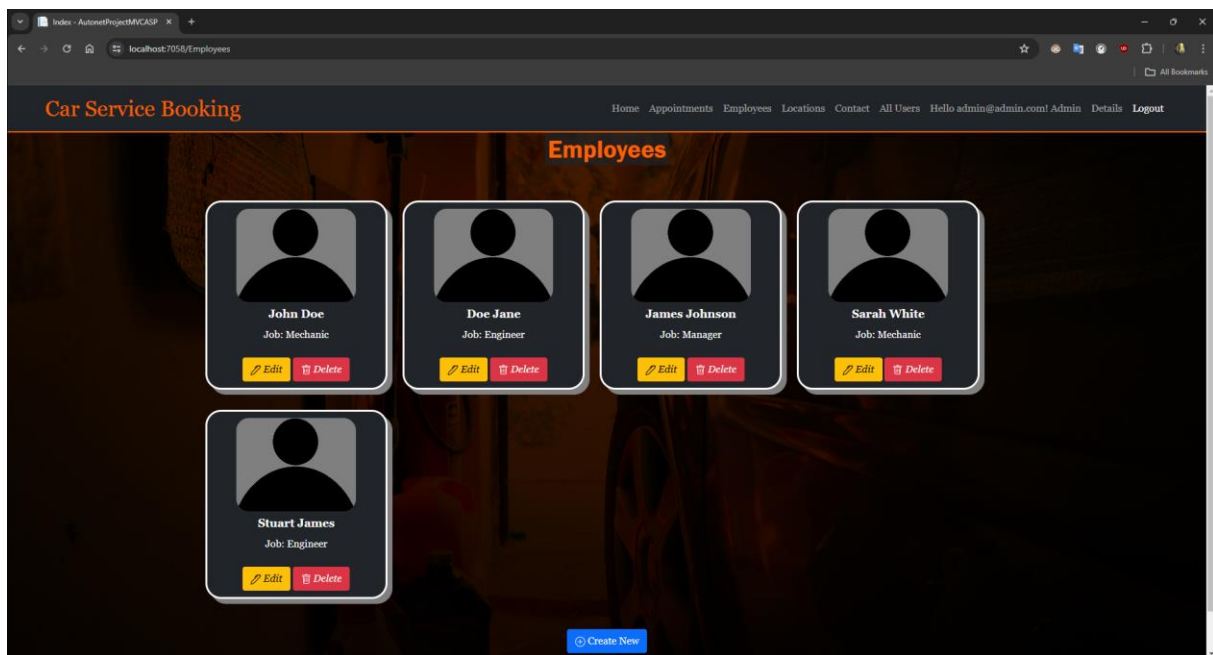


18. ábra: Dolgozók felülete

Itt megnézhetjük a weboldalra regisztrált dolgozók nevét, munkakörét és képét (lásd 17. ábra).

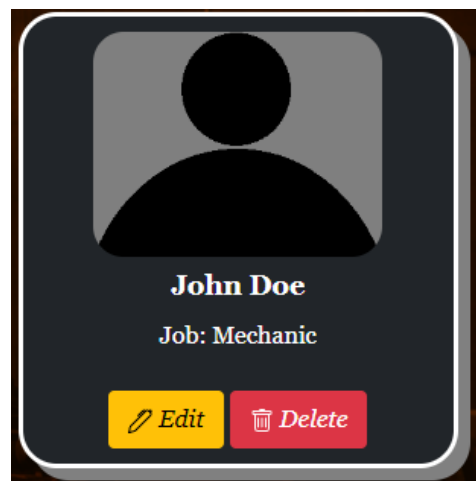


17. ábra: Egy dolgozó kártyája



20. ábra Dolgozók adminisztrátor felülete

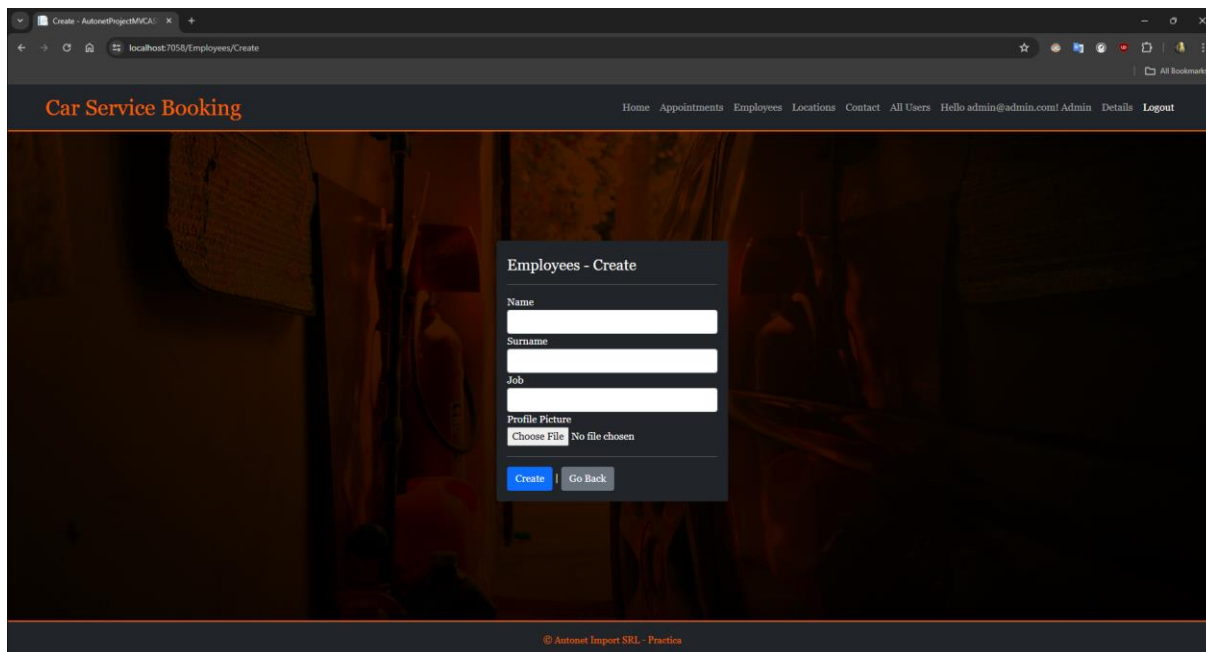
Admin és Főnök jogosultsággal rendelkező felhasználóknak ez az egyszerű felület jóval több funkcionalitást nyer (lásd 20. ábra): Innen képesek lesznek dolgozókat létrehozni, módosítani és törölni (lásd 19. ábra).



19. ábra: Egy dolgozó adminisztrátor módban

2.7.2 Dolgozó létrehozása és módosítása – Speciális jogosultságú funkció

Az oldal alján levő Create New gombra kattintva a következő oldalra kerülünk (lásd 21. ábra):

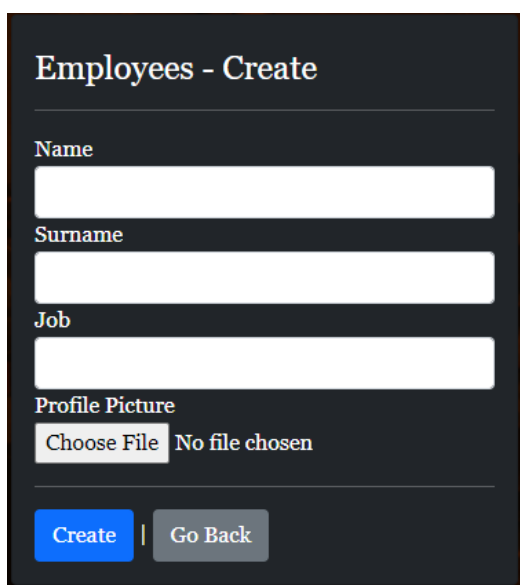


The screenshot shows a web browser window with the URL `localhost:7050/Employees/Create`. The page has a dark theme and a header with the text "Car Service Booking" and navigation links: Home, Appointments, Employees, Locations, Contact, All Users, Hello admin@admin.com, Admin, Details, and Logout. The main content area features a form titled "Employees - Create" with the following fields: Name, Surname, Job, and Profile Picture. The Profile Picture field includes a "Choose File" button and the text "No file chosen". At the bottom of the form are two buttons: "Create" (blue) and "Go Back" (grey). The background of the page shows a blurred image of a car interior.

23. ábra: Dolgozó készítő felület

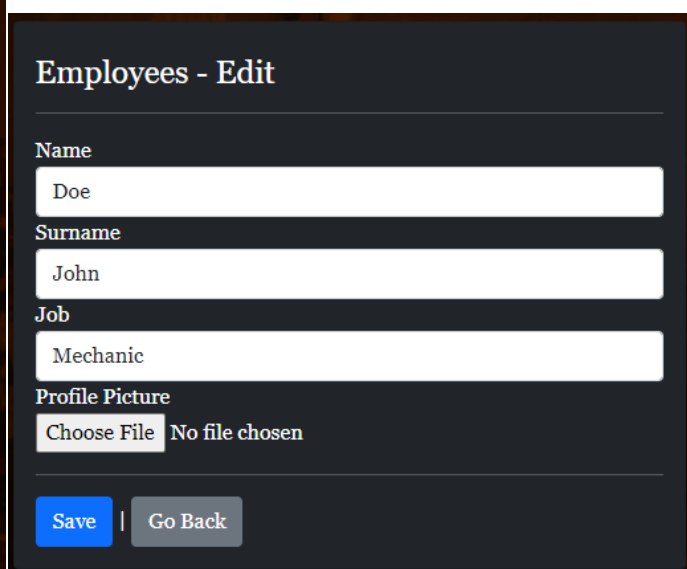
Itt megadhatjuk a dolgozó nevét, munkakörét és képét, a Create gombra kattintva létrehozzuk (lásd 23. ábra).

Viszonylag hasonló felületre kerülünk a dolgozók kártyáin levő Edit gombra kattintva is, itt a fő különbség az, hogy a név és a munkakör az eddigi információk alapján már adott (22. ábra).



This is a detailed view of the "Employees - Create" form. It has a dark background with white text and input fields. The fields are labeled "Name", "Surname", "Job", and "Profile Picture". The "Profile Picture" field has a "Choose File" button and the text "No file chosen". At the bottom, there are two buttons: "Create" (blue) and "Go Back" (grey).

1. ábra: Dolgozó készítési kártya

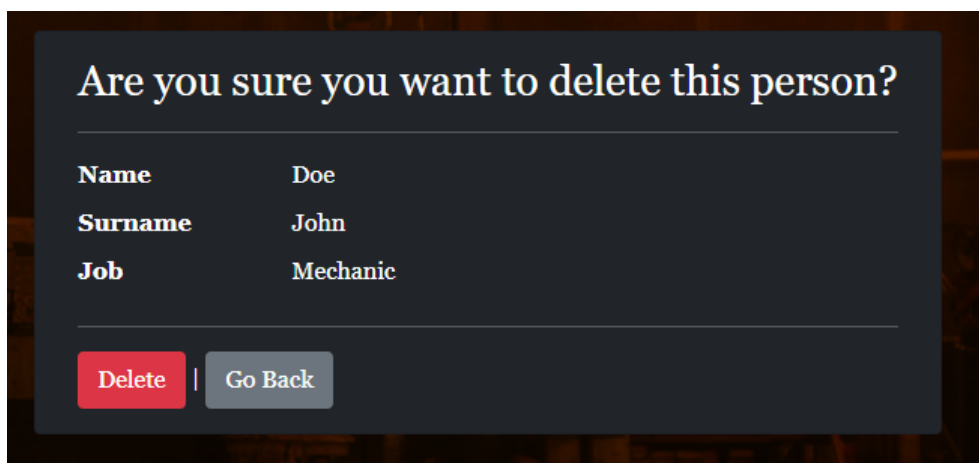


This is a detailed view of the "Employees - Edit" form. It has a dark background with white text and input fields. The fields are labeled "Name", "Surname", "Job", and "Profile Picture". The "Name" field contains the text "Doe", the "Surname" field contains "John", and the "Job" field contains "Mechanic". The "Profile Picture" field has a "Choose File" button and the text "No file chosen". At the bottom, there are two buttons: "Save" (blue) and "Go Back" (grey).

22. ábra: Dolgozó módosítási kártya

2.7.3 Dolgozók törlése – Speciális jogosultságú funkció

A dolgozói kártyán levő törlés gombra kattintva a dolgozóhoz tartozó összegzőfelületre kerülünk, ahol, hasonlóan a foglалásokhoz, átnézhetjük az illető adatait, valamint törölhetjük vagy visszamehetünk (lásd 24. ábra).

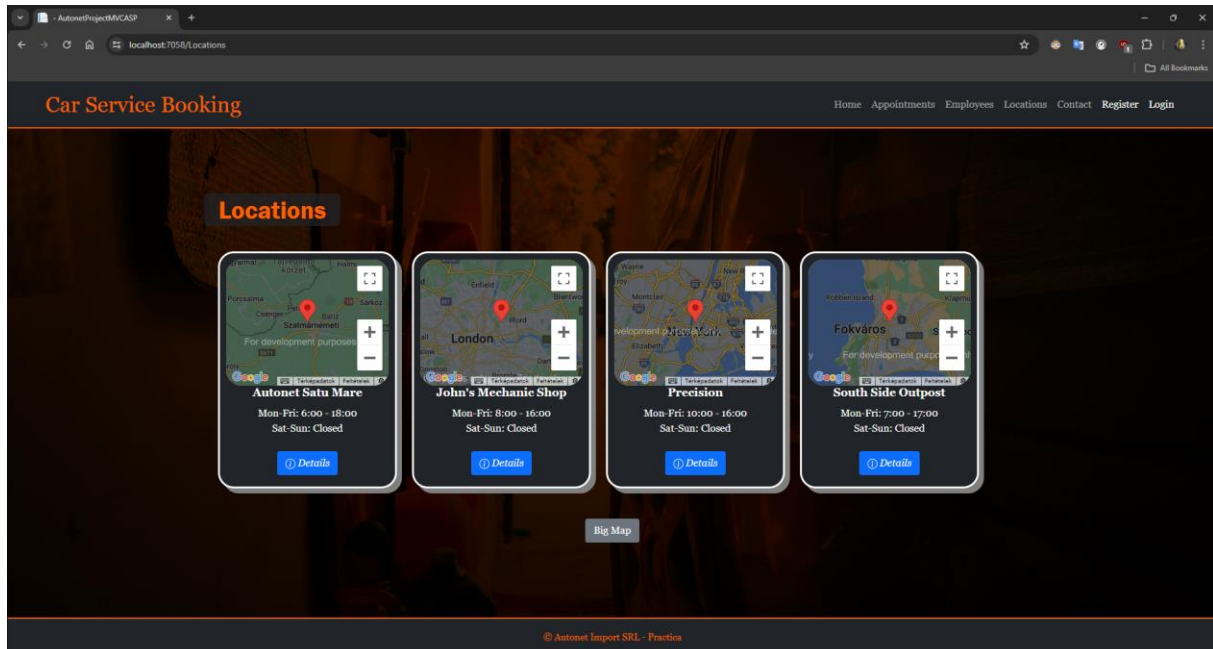


24. ábra: Dolgozó törlési kártya

2.7 Helyszínek

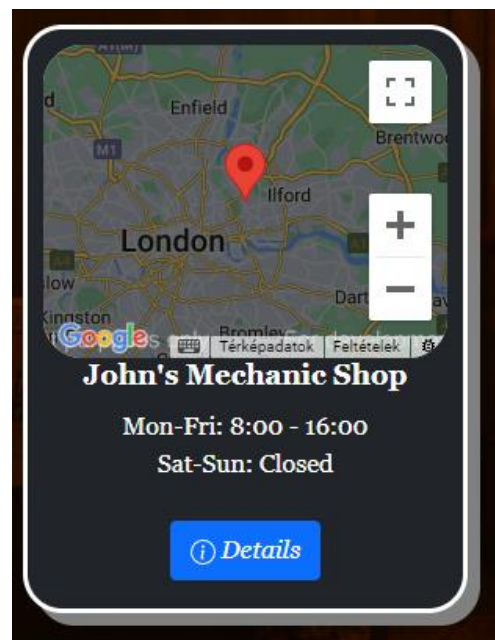
2.7.1 Főoldal

A „Locations” menüpontra kattintva a következő oldal fogad minket (lásd 25. ábra):



25. ábra: A Helyszínek felület

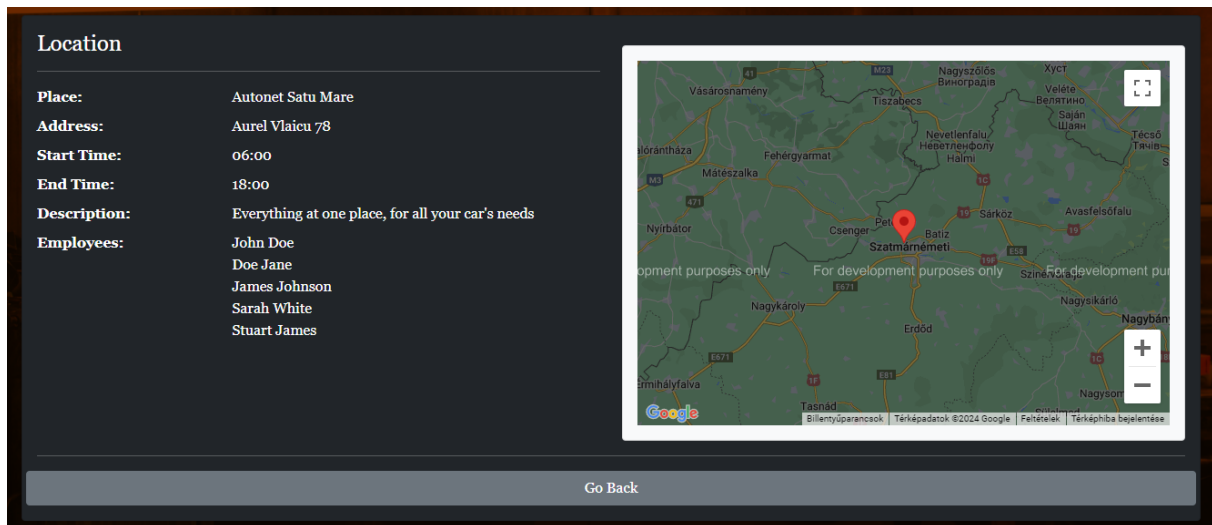
Itt minden kártyán megtekinthetjük egy adott helyszín elsődleges adatait (lásd 26. ábra): a térképen található pozícióját, valamint azt, hogy milyen időpontok között fogad személyeket. A térkép mozgatható, így könnyebben pontosíthatjuk, hogy merre található az adott műhely.



26. ábra: Egy helyszín kártyája

2.7.2 Helyszín részletei

A kártyán levő kék Details gombra kattintva megtekinthetjük az adott helyszín részletes bemutatását (lásd 27. ábra):

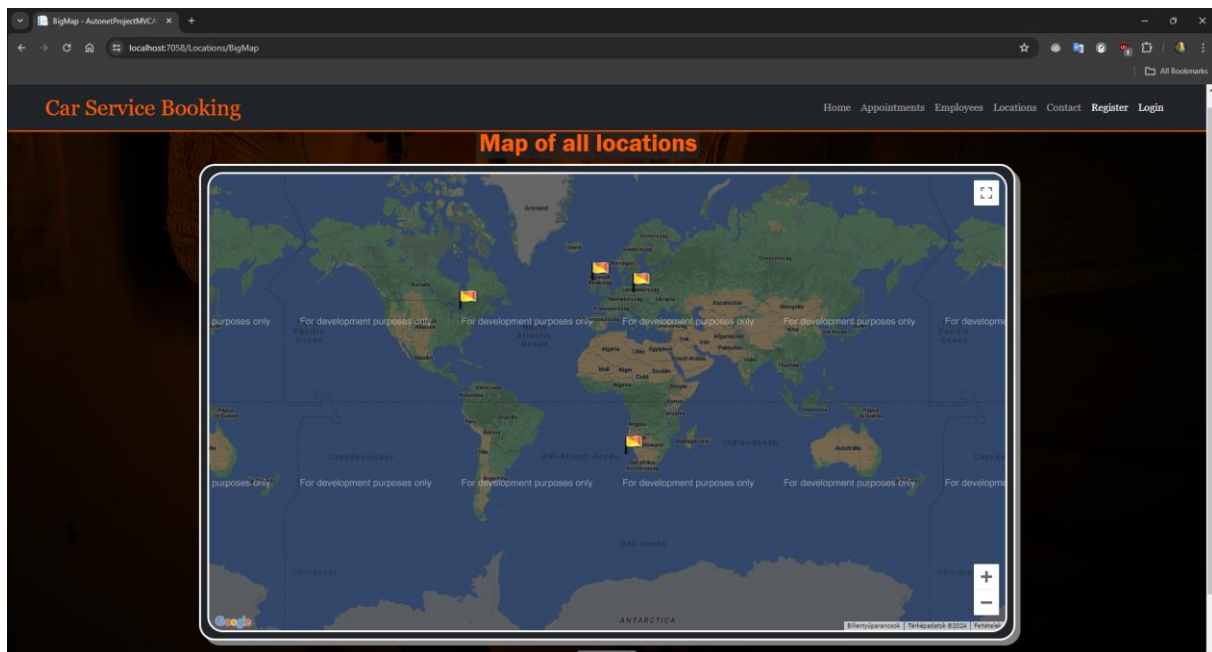


27. ábra: Helyszín részletei kártya

Itt megtekinthetjük a hely nevét és pontos címét, a nyitvatartását, egy rövid leírást, valamint az itt dolgozó személyek listáját.

2.7.3 Nagy térképes megtekintés

Az oldal alján található Big Map gombra kattintva a következő oldalra kerülünk (lásd 28. ábra):



28. ábra: Nagy térkép felület

Ezen az oldalon egy nagy térképen megtekinthetjük az oldalon megjelenő összes helyszín pozícióját, mint apró zászlócskák.

A kurzort a zászlócskákra helyezve további információkat tudhatunk meg, rájuk klikkelve pedig a fent már említett „Részletek” oldalra kerülünk (lásd 29. ábra).

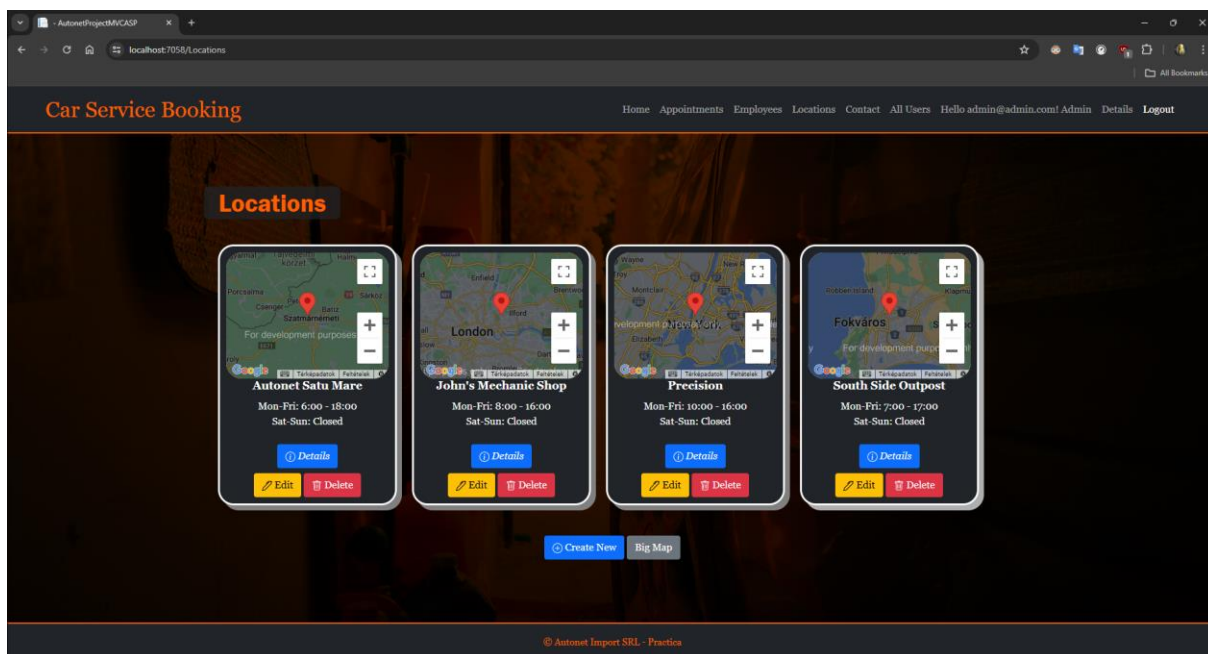
A térkép egy általános Google térkép, így ezen is nagyíthatunk a nagyobb pontosság érdekében.



29. ábra: Egy helyszín a nagy térképen

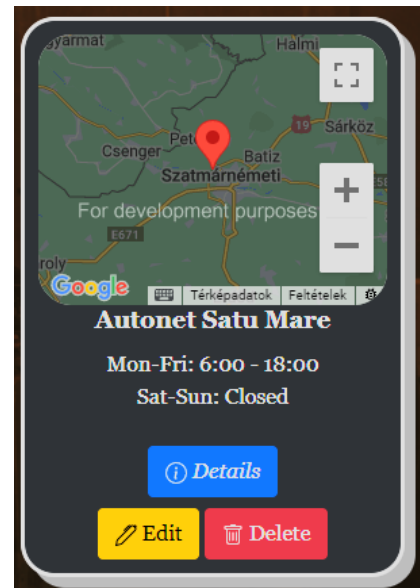
2.7.4 Helyszín menedzsment – Speciális jogosultságú funkció

Admin és Tervező jogokkal rendelkező fiókokban az oldal kissé átalakul (lásd 30. ábra):



30. ábra: Helyszínek adminisztrátor felület

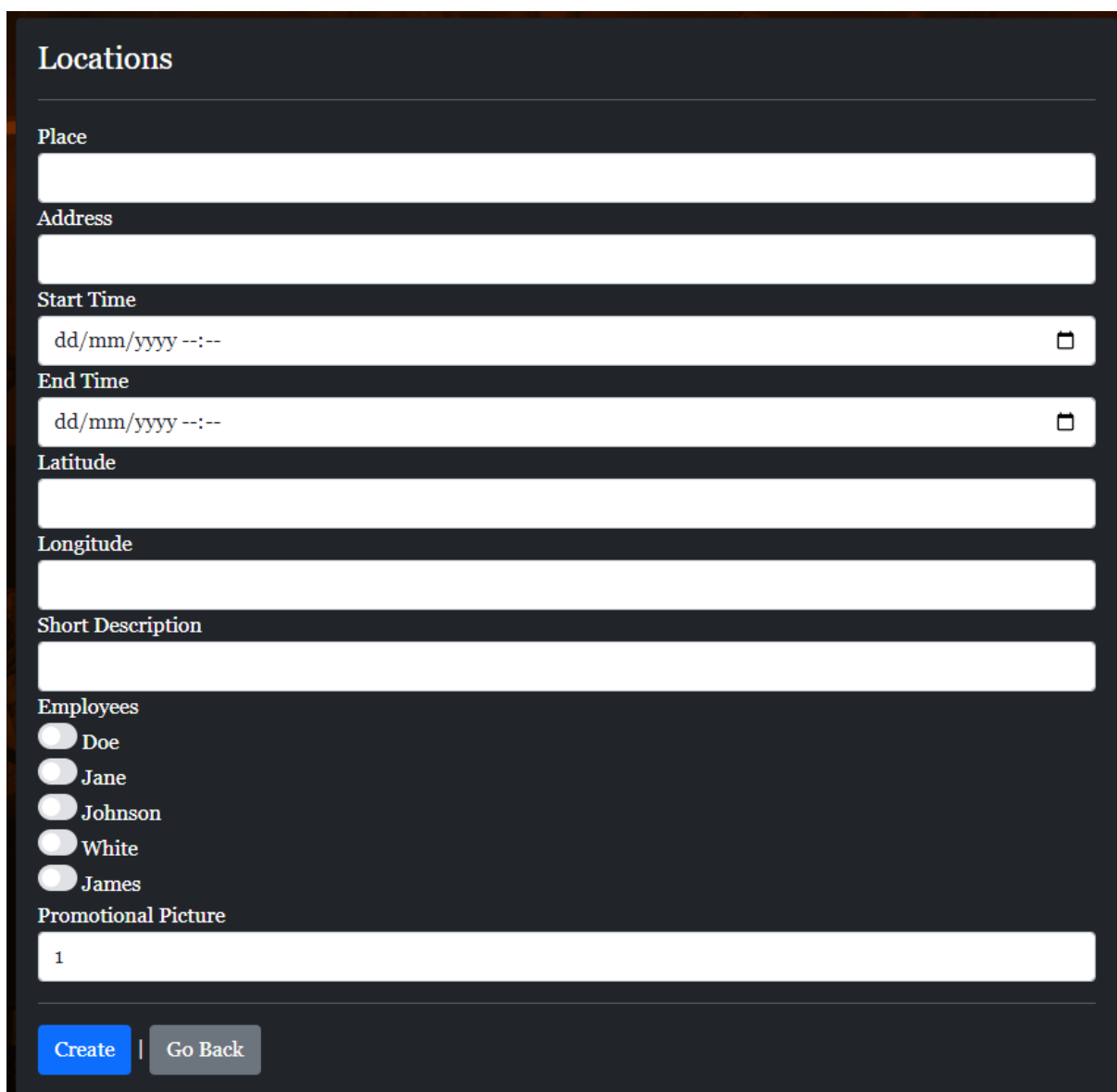
A kártyákon megjelennek az Edit és Delete gombok, valamint a képernyő alján a Create New gomb, amikkel módosíthatunk, törölhetünk helyszíneket, valamint új helyszíneket adhatunk az alkalmazáshoz (lásd 31. ábra).



31. ábra: Egy helyszín adminisztrátor módban

2.7.5 Helyszín létrehozása – Speciális jogosultságú funkció

A Create New gombra kattintva a helyszín létrehozó felületre kerülünk, ahol a következő kártya fogad minket (lásd 32. ábra):



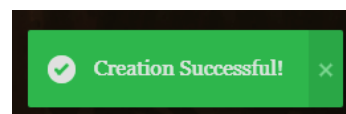
The screenshot shows a 'Locations' form with the following fields and controls:

- Place:** A text input field.
- Address:** A text input field.
- Start Time:** A date and time picker with a calendar icon.
- End Time:** A date and time picker with a calendar icon.
- Latitude:** A text input field.
- Longitude:** A text input field.
- Short Description:** A text input field.
- Employees:** A list of toggle switches for 'Doe', 'Jane', 'Johnson', 'White', and 'James'.
- Promotional Picture:** A text input field containing the number '1'.
- Buttons:** A blue 'Create' button and a grey 'Go Back' button.

32. ábra: Helyszín készítő kártya

Itt megadhatjuk a helyszín nevét, címét, a nyitvatartását (amiből csak az órát vesszük figyelembe az egyszerűség végett), a hely koordinátáit, a rövid leírást, bejelölhetjük, hogy mely dolgozók dolgoznak itt, valamint a Contact felület díszképeihez egy számot.

A sikert egy értesítés nyugtázza (lásd 33. ábra).



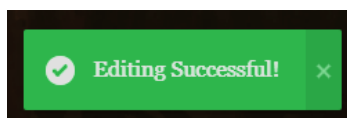
33. ábra: Sikeres létrehozás jelzése

2.7.6 Helyszín módosítása – Speciális jogosultságú funkció

A módosítás gombra kattintva egy igen hasonló felületre kerülünk (lásd 34. ábra):

Ez nagyon hasonlít a helyszín készítő felülethez, a fő különbség az, hogy a dolgozók kivételével, minden az eddigi adatok alapján előre be van írva, valamint, hogy a díszkép változtatására már nincs lehetőség.

A sikert egy értesítés nyugtázza (lásd 35. ábra).

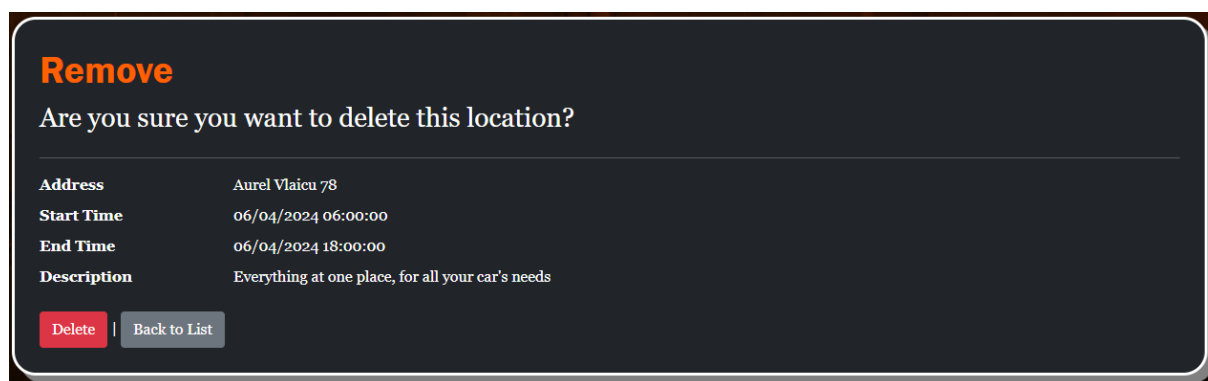


35. ábra: Sikeres módosítás jelzése

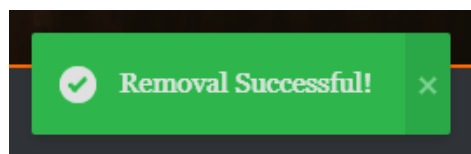
34. ábra: Helyszín módosítás kártya

2.7.7 Helyszín törlése – Speciális jogosultságú funkció

A kártyán levő törlés gombra kattintva egy ellenőrző felületre kerülünk, ahol a fő adatok átnézése után törölhetjük az elemet, vagy visszatérhetünk (lásd 36. ábra). Itt is a sikeres műveletet értesítés követ (lásd 37. ábra). Meg kell jegyeznünk, hogy a fent említett összes jelzés minden egyes létrehozási, törlési és módosítási műveletnél is megjelenik, nem csak a helyszínek műveleteinél.



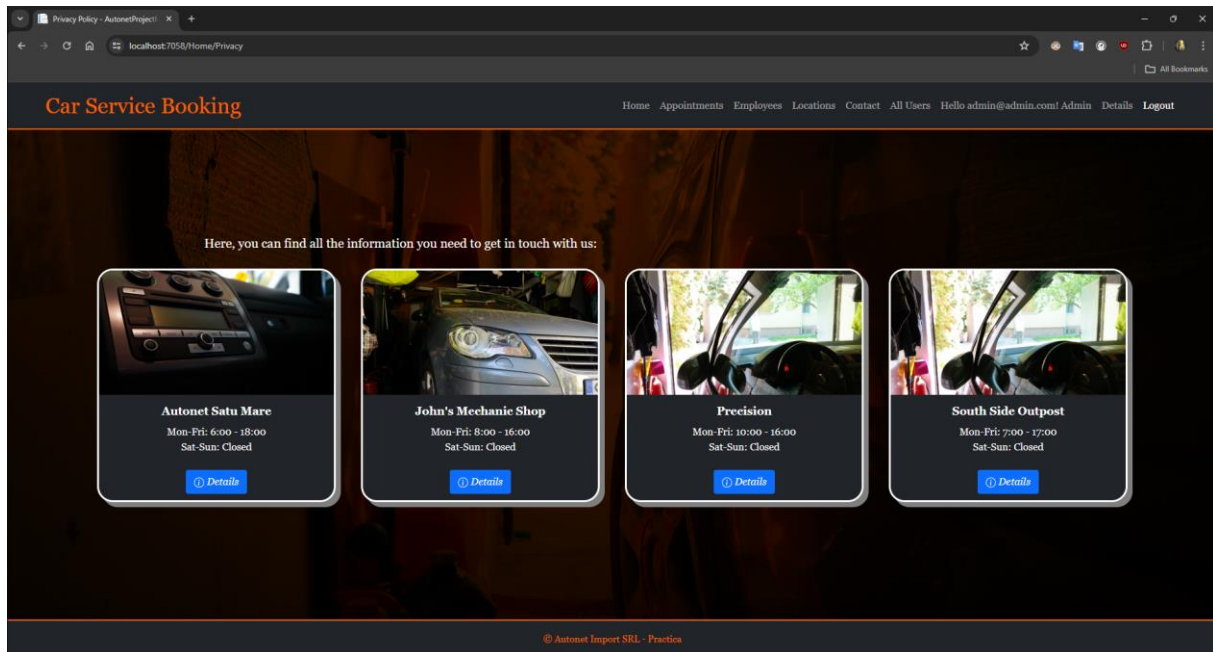
36. ábra: Helyszín törlés kártya



37. ábra: Sikeres törlés jelzés

2.8 Elérhetőségek

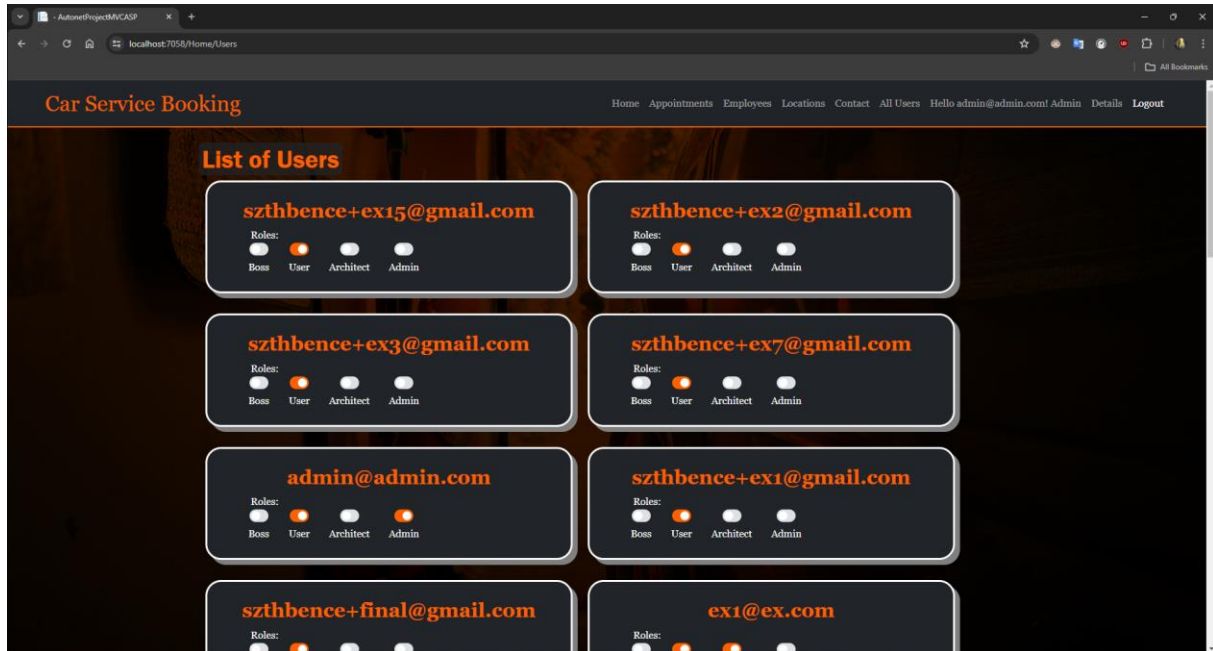
A Contact menüpontra kattintva egy képes oldalra kerülünk (lásd 38. ábra), ahol a Helyszín-leíróhoz hasonló információkat találunk. Itt is bővebb információkat érhetünk el a Details (Részletek) gombokra kattintva.



38. ábra: Részletek felület

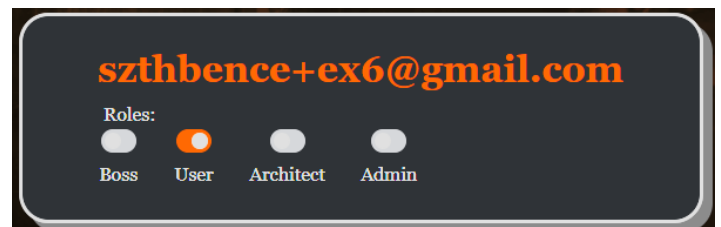
2.9 Fiókkezelő oldal – Speciális jogosultságú funkció

Admin felhasználóknak rendelkezésére áll egy All Users menüpont is a fejlécben, amire kattintva a fiókokhoz tartozó jogosultságokat tudjuk állítani (lásd 39. ábra).

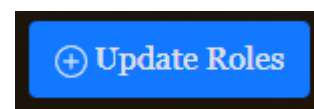


39. ábra: Fiókkezelő felület

Itt minden fiók rendelkezik egy kártyával (lásd 41. ábra), ami alatt ki vannak választva a neki megadott szerepkörök (lásd a Felhasználó típusok szekcióban – 2.4). A változtatásokat az oldal alján található Update Roles gombbal véglegesítjük (lásd 40. ábra).



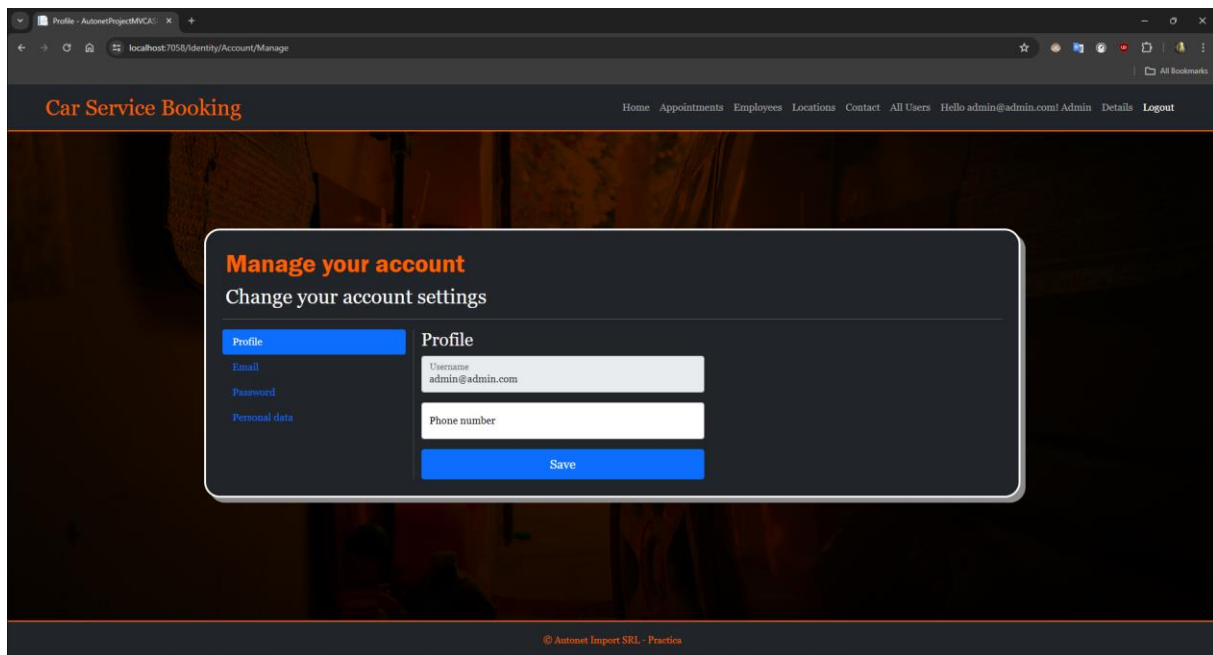
41. ábra: Egy fiók kártyája



40. ábra: Változás mentés gomb

2.10 Fiók adatok és menedzsment

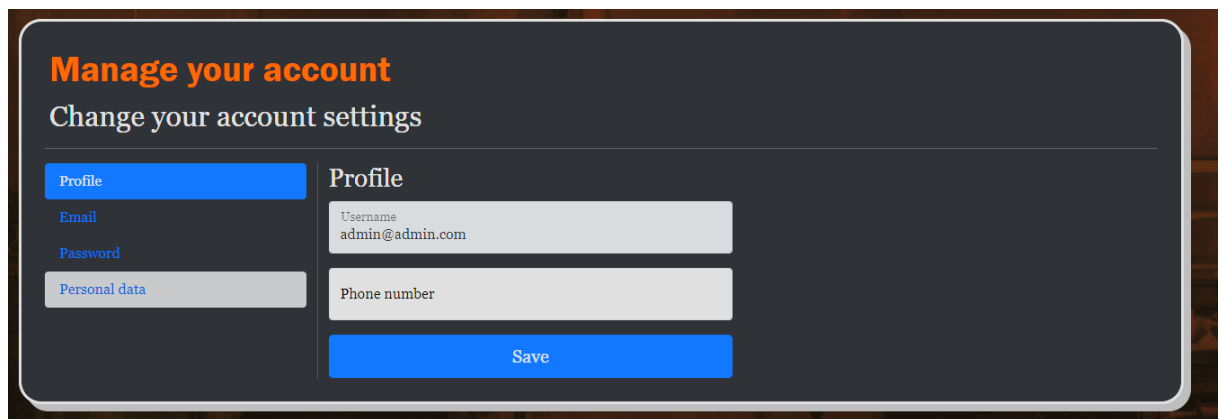
A köszönőszövegre kattintva a fiókmenedzsment részlegre kerülünk (lásd 42. ábra). Itt további információkat adhatunk a fiókunkhoz, valamint meglévő adatokat változtathatunk. A navigálás négy oldalon történik, ezek között a kártya bal oldalán lévő menüpontokkal váltogathatunk.



42. ábra: Fiók menedzsment felület

A menüpontok a következők:

- Profile – Itt új telefonszámot csatolhatunk a fiókhoz, valamint látjuk a fiók nevét, azaz az email címét (lásd 43. ábra).



43. ábra: Menedzsment profil kártya

- Email – Új email cím megadásának felülete (lásd 44. ábra).

Manage your account
Change your account settings

Profile
Email
Password
Personal data

Manage Email

Email
admin@admin.com ✓

New email
admin@admin.com

Change email

44. ábra: Menedzsment email kártya

- Password – Új jelszó megadásának felülete. Ennek használatához ismernünk kell a fiókhoz tartozó jelenlegi jelszót is (lásd 45. ábra).

Manage your account
Change your account settings

Profile
Email
Password
Personal data

Change password

Current password

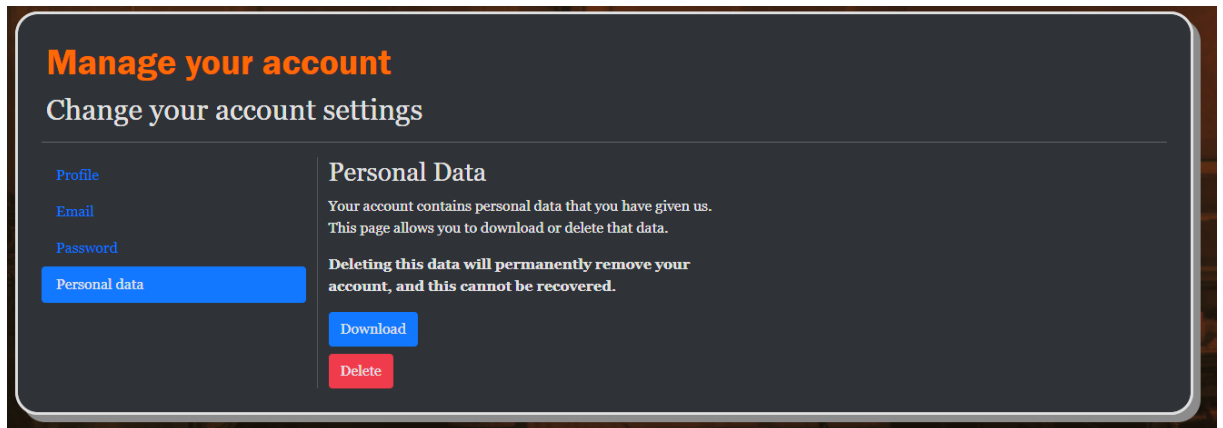
New password

Confirm new password

Update password

45. ábra: Menedzsment jelszó kártya

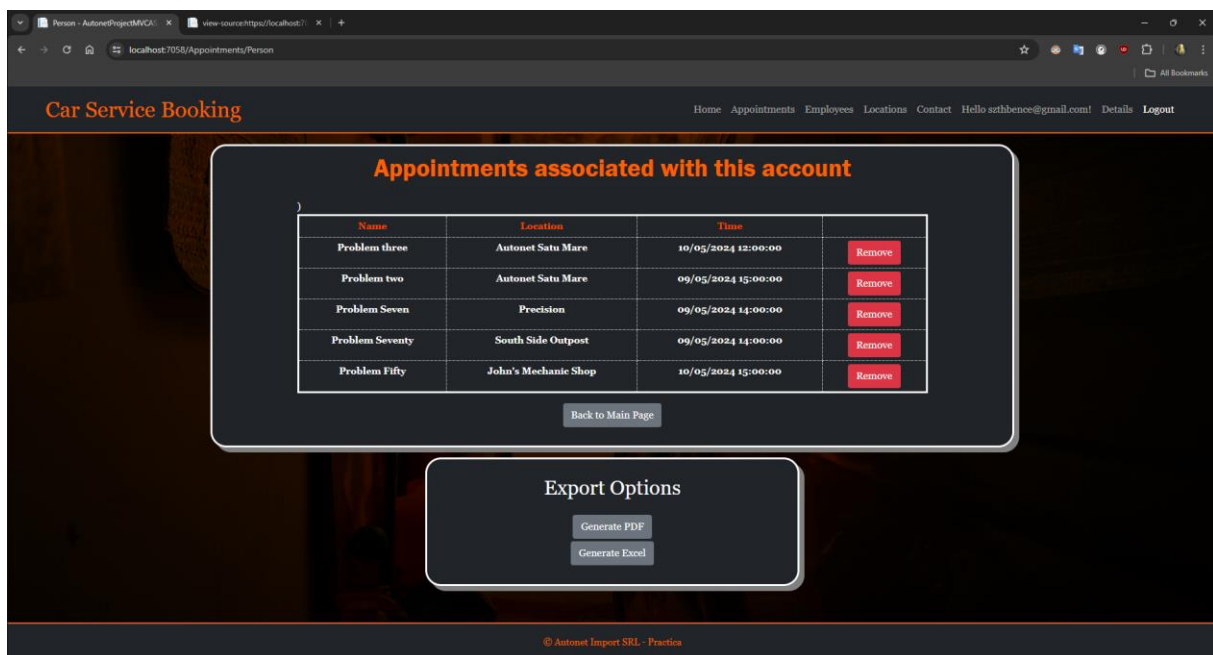
- **Personal Data** – Ezen az oldalon letölthetjük a fiókunkhoz tartozó személyes információkat, valamint törölhetjük a fiókunkat (lásd 46. ábra).



46. ábra: Menedzsment személyes adatok kártya

2.11 Fiók foglalási részletek

A fejléc Details gombjára kattintva a következő oldalra kerülünk (lásd 47. ábra):



47. ábra: Fiók foglalási részletek felület

Az oldal két fő részből áll:

- A felső kártyán megtekinthetjük a saját foglalásaink nevét, helyét és idejét, valamint lehetőségünk van a foglalás törlésére is (lásd 48. ábra). Ez a feljebb ismertetett törlési összegző lapra visz minket (lásd Foglalás törlése – 2.6.5). Ha nincs foglalásunk, erről üzenetet kapunk (lásd 49. ábra).

| Appointments associated with this account | | | |
|---|----------------------|---------------------|--------|
| Name | Location | Time | |
| Problem one | Autonet Satu Mare | 09/04/2024 15:00:00 | Remove |
| Problem two | Autonet Satu Mare | 09/04/2024 16:00:00 | Remove |
| Problem three | John's Mechanic Shop | 10/04/2024 12:00:00 | Remove |
| Problem Seventy | Autonet Satu Mare | 11/04/2024 14:00:00 | Remove |
| Problem Forty-four | Precision | 11/04/2024 14:00:00 | Remove |

Back to Main Page

48. ábra: Fiók foglalási táblázat kártya

| Name | Location | Time |
|---|----------|------|
| There are no appointments associated with this account! | | |

49. ábra: Üres táblázat jelző

- Az alsó kártyán lehetőségünk van a saját fiókunkhoz tartozó foglalások listájának letöltésére pdf (lásd 51. ábra) vagy Excel (lásd 52. ábra) formában (lásd 50. ábra).



50. ábra: Letöltési lehetőségek kártya

A felhasználóhoz tartozó pdf (lásd 51. ábra):

| admin@admin.com's appointments in the database | | | | | |
|--|--------------------|---------------------|----------------------|-----------------|-------------|
| Id | Appointment Name | Time of Appointment | Location | User | Employee |
| 39 | Problem one | 09/04/2024 15:00:00 | Autonet Satu Mare | admin@admin.com | Doe |
| 40 | Problem two | 09/04/2024 16:00:00 | Autonet Satu Mare | admin@admin.com | Doe |
| 41 | Problem three | 10/04/2024 12:00:00 | John's Mechanic Shop | admin@admin.com | Doe |
| 43 | Problem Seventy | 11/04/2024 14:00:00 | Autonet Satu Mare | admin@admin.com | Doe |
| 44 | Problem Forty-four | 11/04/2024 14:00:00 | Precision | admin@admin.com | No Employee |

51. ábra: Saját felhasználó pdf

A felhasználó Excel táblázata (lásd 52. ábra):

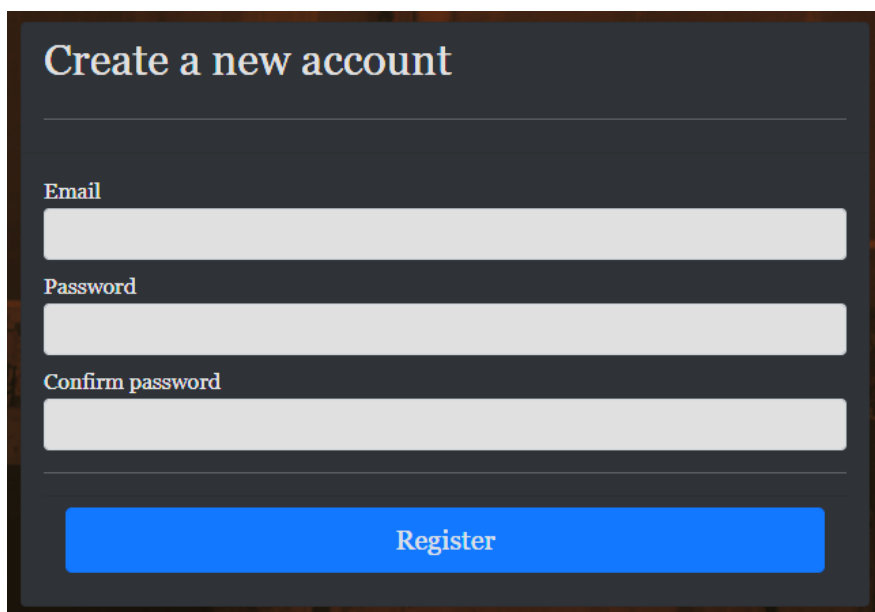
| | A | B | C | D | E | F | G |
|---|----|-----------|----------|-----------|----------|----------|-------|
| 1 | Id | Name | Time | Location | UserId | Employee | JobId |
| 2 | 39 | Problem c | 45391.63 | Autonet S | admin@ai | 5 | 46 |
| 3 | 40 | Problem t | 45391.67 | Autonet S | admin@ai | 5 | 47 |
| 4 | 41 | Problem t | 45392.5 | John's Me | admin@ai | 5 | 48 |
| 5 | 43 | Problem S | 45393.58 | Autonet S | admin@ai | 5 | 50 |
| 6 | 44 | Problem F | 45393.58 | Precision | admin@ai | -1 | 51 |

52. ábra: Saját felhasználó táblázat

2.12 Regisztráció és Bejelentkezés

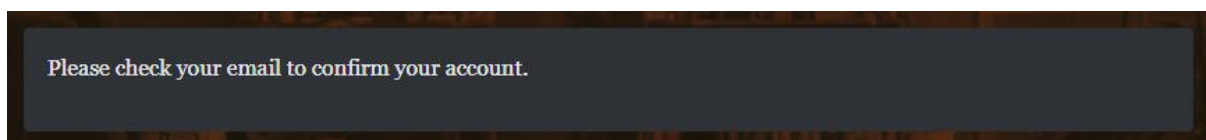
2.12.1 Regisztráció

Kijelentkezett állapotban a fejléc Register gombjára kattintva a regisztrációs felületre kerülünk (lásd 53. ábra). Ilyenkor meg kell adjuk az email címünket, valamint a kívánt jelszót, amit meg is kell erősítenünk ennek újabb megadásával.

A screenshot of a web registration form titled "Create a new account". The form is set against a dark background. It contains three input fields: "Email", "Password", and "Confirm password", each with a light gray placeholder. Below these fields is a prominent blue button labeled "Register".

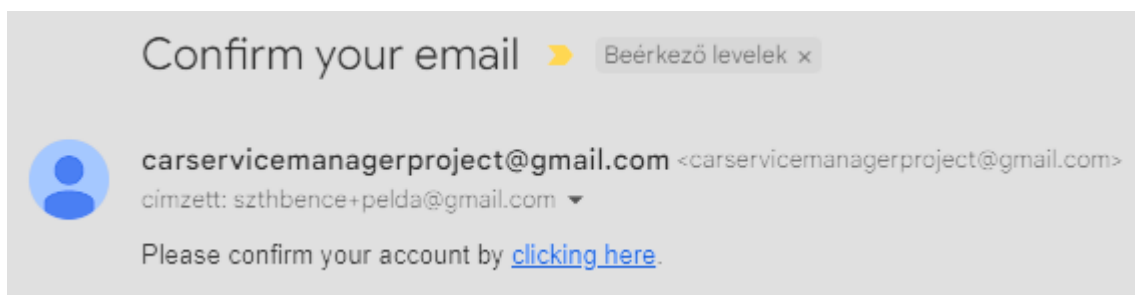
53. ábra: Regisztráció kártya

A regisztráció után egy újabb oldalra kerülünk, ahol a következő üzenet fogad minket (54. ábra):



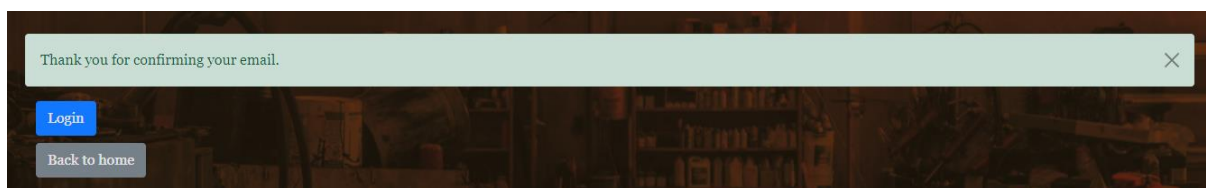
54. ábra: Regisztráció utáni üzenet

A rendszer regisztráció után kiküld egy megerősítő linket tartalmazó emailt (lásd 55. ábra).



55. ábra: Regisztrációs email megerősítő linkel

A linkre való klikkeléssel aktiváljuk a regisztrált fiókot, ezt tudatja is velünk a rendszer a következő oldalon (lásd 56. ábra).



56. ábra: Aktiválás utáni üzenet

Ez után a fiók aktiválva lett, be lehet vele jelentkezni.

2.12.2 Bejelentkezés

A Login gombra kattintva a következő bejelentkező felület fogad minket (lásd 57. ábra):

A screenshot of a login form. The form has a dark grey background with white text. At the top, it says "Log in" in a large, serif font. Below this, there are two input fields: "Email" and "Password". The "Email" field is a light grey rectangle. The "Password" field is a light grey rectangle with a small eye icon on the right. Below the password field, there is a checkbox labeled "Remember me?". Below the checkbox, there is a blue "Log in" button. At the bottom of the form, there are three links: "Register as a new user", "Forgot your password?", and "Resend email confirmation".

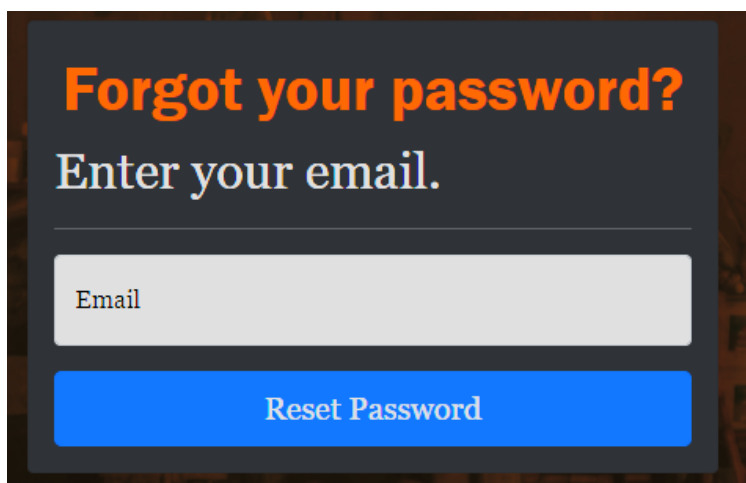
57. ábra: Bejelentkezési kártya

Itt az email címünk és az ehhez tartozó jelszó megadásával beléphetünk a fiókunkba. Lehetőség van a fiók megjegyzésére, hogy legközelebb automatikusan belépjünk, valamint további segítő funkciókat is igénybe vehetünk, mint az elfelejtett jelszó visszaszerzése és a megerősítő email újra küldése.

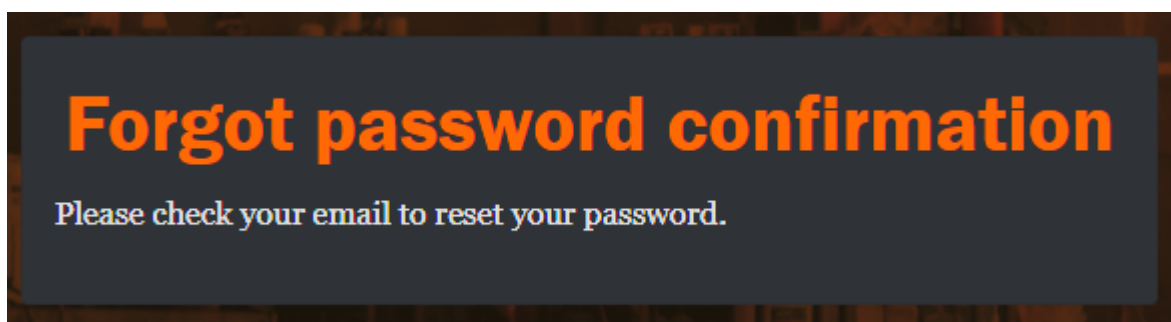
2.12.3 Elfelejtett jelszó

A “Forgot your password” linkre klikkelve egy új felületre kerülünk (lásd 58. ábra), ahol megadhatjuk annak a fióknak az email címét, aminek elfelejtettük a jelszavát.

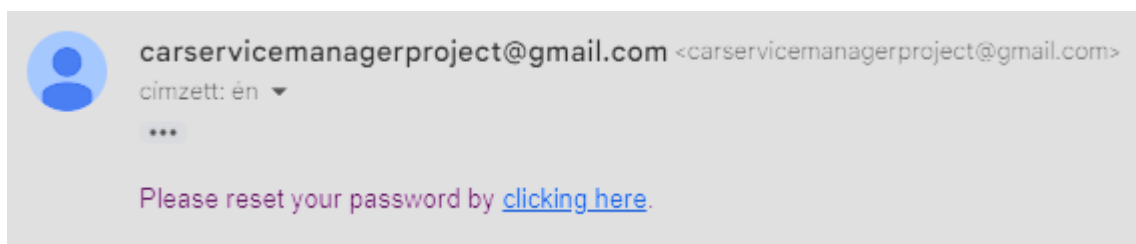
A Reset Password gombot megnyomva az alkalmazás elküld egy emailt (lásd 60. ábra), amiben egy visszaállító oldalra mutató linket találunk. Emellett egy új oldalra is irányít minket a rendszer (lásd 59. ábra).

A screenshot of a web form titled "Forgot your password?" in orange text. Below the title, it says "Enter your email." in white text. There is a light gray input field with the placeholder text "Email". Below the input field is a blue button with the text "Reset Password" in white.

58. ábra: Elfelejtett jelszó kártya



59. ábra: Email megadása utáni üzenet



60. ábra: Jelszó visszaállító email

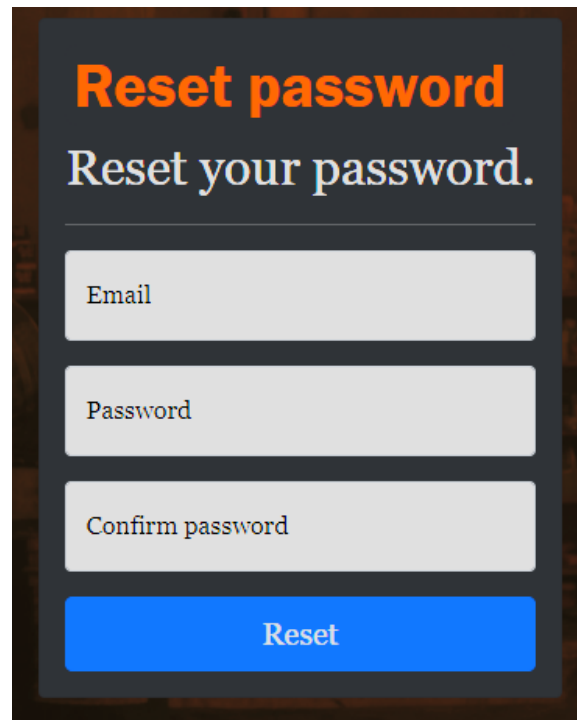
A linkre klikkelés után egy, a regisztrációs felülethez hasonló felületre kerülünk, ahol megadhatjuk az új adatainkat (lásd 61. ábra). Ezt csak egyszer használhatjuk fel, és csak a saját fiókunk jelszavát tudjuk megváltoztatni, másképp hibát ad a rendszer (lásd 62. ábra).

- **Invalid token.**

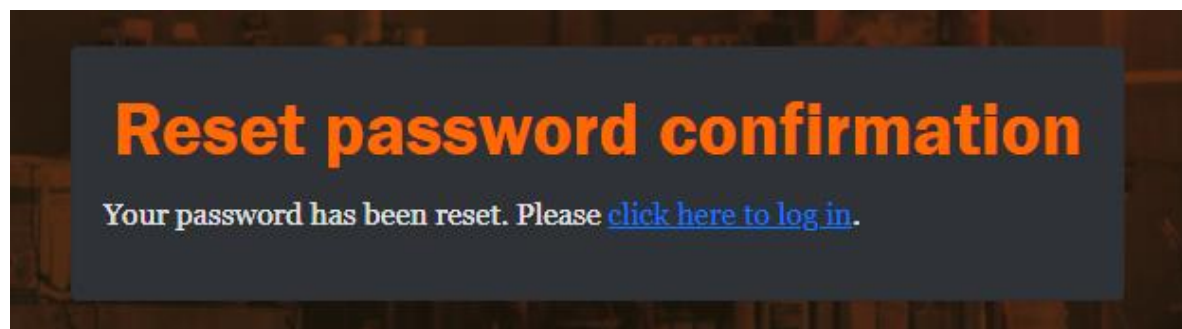
62. ábra: Többszörös használat hiba

Természetesen az új jelszónak is meg kell felelnie bizonyos elvárásoknak a hosszát és tartalmát illetően.

Ha mindent sikeresen végrehajtottunk, akkor a következő oldal jelzi, hogy minden sikeresen végbement (lásd 63. ábra):



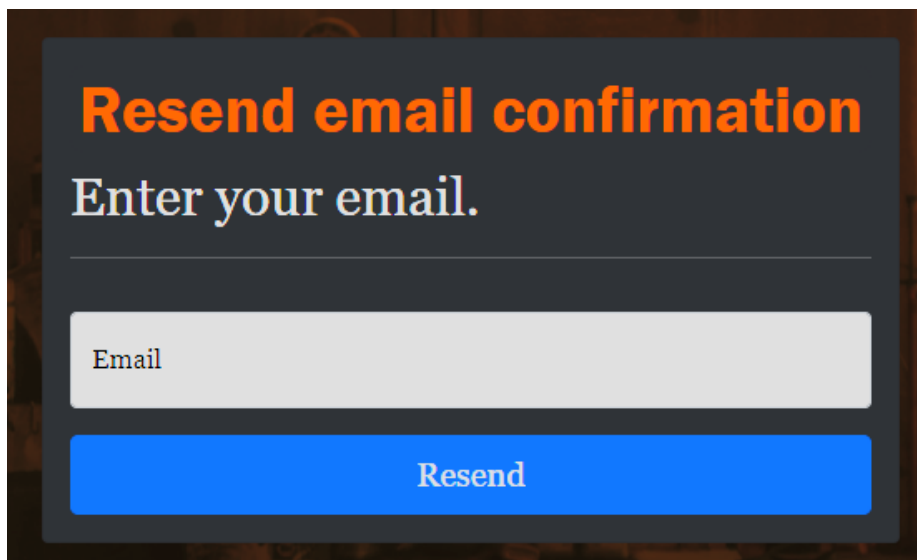
61. ábra: Új jelszó megadása kártya



63. ábra: Sikeres jelszóváltoztatás üzenet

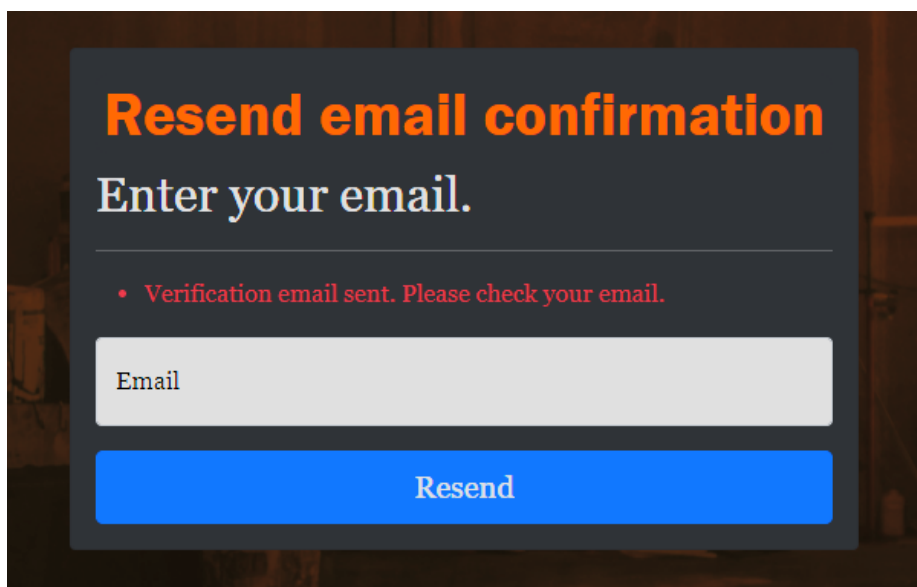
2.12.4 Megerősítő email újra küldése

Ha elfelejtettük aktiválni fiókunkat, a rendszer nem enged minket belépni, ameddig ezt meg nem tettük. Ha elvesztettük a megerősítő emailt, a „Resend email confirmation” linkre kattintva új email igényelhetünk (lásd 64. ábra).

A dark-themed modal window with a title bar. The title is "Resend email confirmation" in orange. Below it is the instruction "Enter your email." in white. There is a light gray input field with the placeholder text "Email". Below the input field is a blue button with the text "Resend" in white.

64. ábra: Megerősítés újra küldése kártya

Ennek elküldése után a következő státusz-üzenetet kapjuk (lásd 65. ábra):

A dark-themed modal window, similar to the previous one. The title is "Resend email confirmation" in orange. Below it is the instruction "Enter your email." in white. A red bullet point message is displayed: "• Verification email sent. Please check your email." Below this message is a light gray input field with the placeholder text "Email". Below the input field is a blue button with the text "Resend" in white.

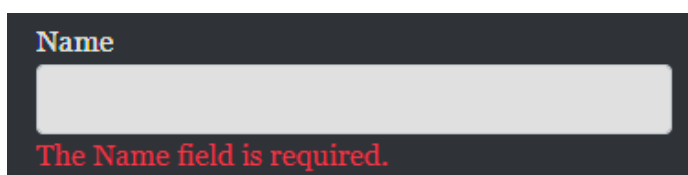
65. ábra: Megerősítés újra küldési kártya sikeres küldés után

Ezután megerősíthetjük a fiókunkat a fent már leírt módon.

2.13 Hiba- és egyéb jelzések

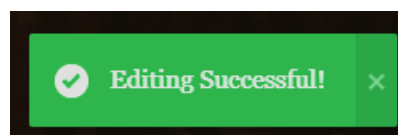
Az alkalmazás használata során rengeteg ellenőrzésbe is belefutunk, amikhez rossz információkat is megadhatunk. A hibás adatok elkerülése érdekében rengeteg ellenőrzéssel láttuk el az oldalt, hogy az adatbázis biztosan helyes adatokkal dolgozzon. Ezek több módon is megjelenhetnek:

- Statikus hibakiírás – Hibás adatok, vagy hiányzó adatok esetén nem enged tovább (lásd 66. ábra).

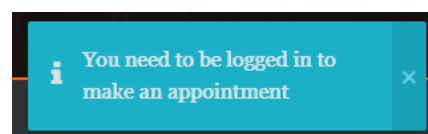


66. ábra: Hiányzó név hiba

- Toast értesítések – Bizonyos műveletek elvégzése után Toast értesítéseket küldünk, amikkel egy művelet hatására utalhatunk (lásd 67. ábra), vagy további információt adhatunk az adott oldal működéséről (lásd 68. ábra).



67. ábra: Sikeres módosítás



68. ábra: Információs ablak, foglalás csak bejelentkezés után lehetséges

A program futása során a következő hibaüzenetekbe futhatunk bele:

2.13.1 Statikus hibák

- Helyszín készítése és módosítása (lásd 1. táblázat).

| | |
|-------------------------------|--|
| The Place field is required | Hiányzó helyszín. |
| The Address field is required | Hiányzó cím. |
| The value is invalid | Értékelhetetlen adat, általában üres. |
| The latitude must be valid | Rossz szélességi kör, 90 alatt kell legyen. |
| The longitude must be valid | Rossz hosszúsági kör, 180 alatt kell legyen. |

1. táblázat: Statikus hibák helyszíneknél

- Dolgozó készítése és módosítása (lásd 2. táblázat).

| | |
|-------------------------------|-------------------------|
| The Name field is required | A vezetéknév mező üres. |
| The Surname field is required | A keresztnév mező üres. |

2. táblázat: Statikus hibák dolgozóknál

2.13.2 Toast értesítések

| | |
|--|--|
| Siker - "Action" successful! | Az adott művelet sikeresen végrehajtódott. |
| Siker – Roles updated successfully | Sikeresen frissítettük a felhasználók jogosultságait. |
| Információ – You need to be logged in to make an appointment | A foglaló táblázatnál járunk, de nem vagyunk még bejelentkezve. |
| Hiba – Name field is required | Foglalásnál nem adtuk meg a foglalás nevét. |
| Hiba – Unexpected Login error | A személyes foglalások oldalon vagyunk fiók nélkül. Ez a jelen állapotban soha nem valósulhat meg. |
| Hiba – "Action" failed! | Általános hiba, általában statikus hibákkal egyszerre jelenik meg. |
| Hiba – Please select an image file | Dolgozó adatainak manipulálásánál nem volt kép megadva. |
| Hiba – Concurrency error occurred! | Több művelet próbálta ugyanazt az elemet elérni, képfeltöltés esetén jelenhet meg. |

3. táblázat: Toast értesítések tartalmai

Fent (lásd 3. táblázat) az „Action” szócska helyett a létrehozó (Creation), módosító (Edit) és törlő (Deletion) műveletek nevei jelenhetnek meg.

3. Fejlesztői dokumentáció

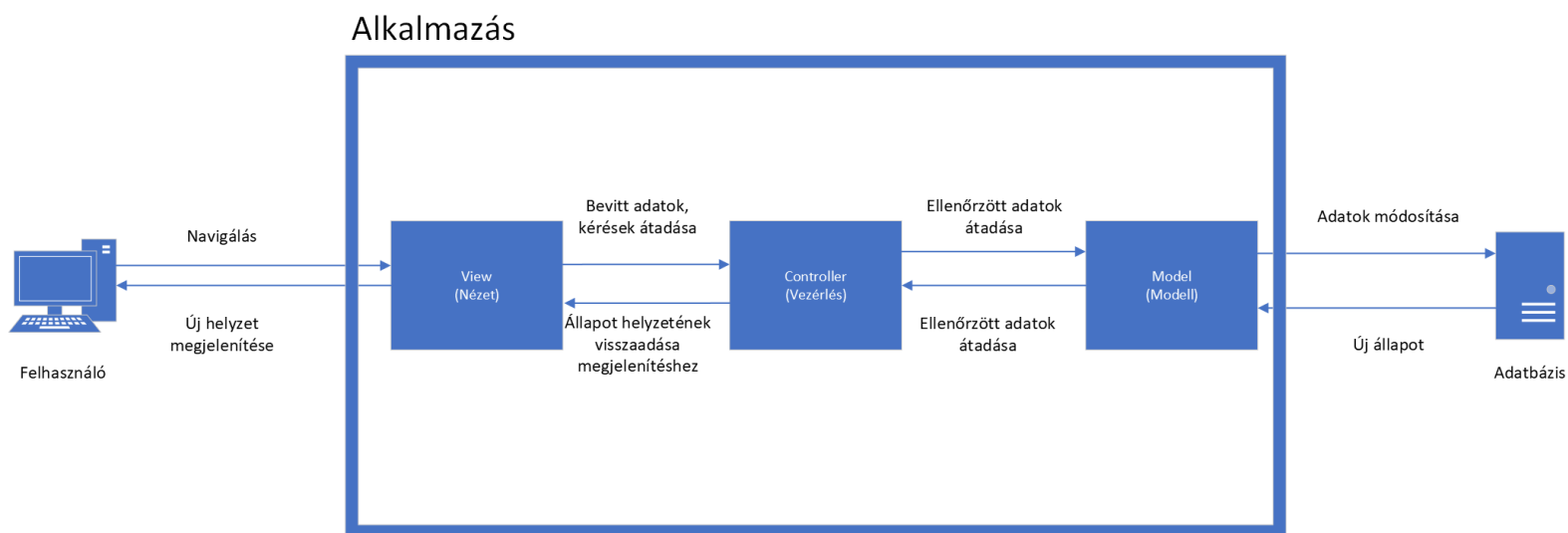
3.1 Áttekintés és Tervezés

A Car Service Manager applikáció egy ASP.NET alkalmazás, alapja a .NET 8.0, aminek nagy részét MVC paradigma [4] szerint működtetem. Alapelvem, hogy a Modell (Adatbázis) és a View (Nézetek) közötti kapcsolatot egy Controller (Irányító) réteg biztosítja, így a működési logika jól elkülönül. Ennek következtében a három réteg feladatkörei a következők:

- **Modell** – Az adatbázissal való kapcsolat rétege. Az alkalmazásunk adatbázisa kód alapú („Code first” módszer), a Modell osztályai alapján képezzük az adatbázisban. Ez azt jelenti, hogy az adatbázis felépítését a modell osztály elemei fogják meghatározni, valamint minden egyes adatelem változtatás után szükségünk lesz az adatok aktualizálására az adatbázisban. Ezek az osztályok nagyrészt csak adatelem jelzőket tartalmaznak, kivétel lehet erre a toString metódus, ami a kiírásban segít, így nem kell minden kiírásban formázni az elemeket.
- **Controller** – Az alkalmazás logikájának lelke. Itt találhatóak az oldalak betöltésére szolgáló logikai elemek. Fő elemek a Get és Post metódusok, a kisegítő metódusok, valamint az adatátvitelhez használt segédosztályok.
- **View** – Ez a réteg biztosítja a megjelenítést. Itt úgy nevezett „Razor Page”-eket találunk, amik kombinálják a HTML és JavaScript nyelvek erősségeit a C# nyelv elemeivel. Ezek az oldalak tartalmazhatnak úgy nevezett „model” elemeket, melyek adatait a kontroller biztosítja, ez alapján jöhet létre a megjelenítés lényegi része, valamint az űrlapok tartalmát is ezeken adjuk vissza a kontrollereknek.

A fentiek mellett szükségünk lesz még egy adatbázisra, ebben az esetben a Microsoft által készített SQL adatbázist használjuk, amit az SQL Server Management Studio 19 [3] (továbbiakban SSMS) programmal ellenőrizhetünk. Jelen állapotban a szerver az alap „localhost” helyen fut, ez az SSMS-ben elérhető a „.” megadásával is.

Az adatbázis mellett természetesen szükséges még egy internetes böngésző is. Fejlesztés alatt az Opera [5] és Google Chrome [6] böngészők voltak használva, így feltételezhetően minden modern Chromium [7] alapú böngészőben hiba nélkül futnia kell az oldalnak.



69. ábra: Az alkalmazás felépítése

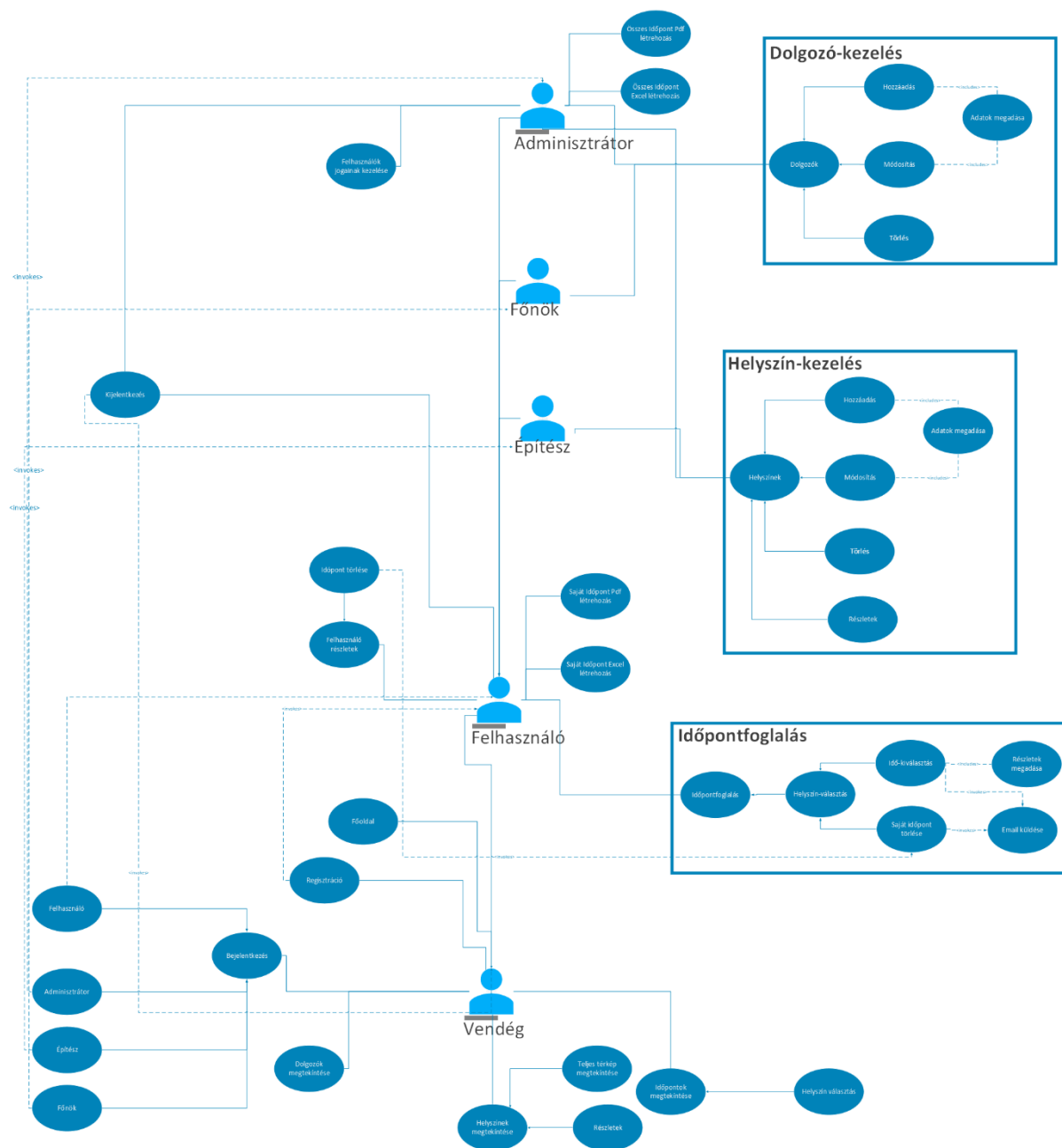
Fent láthatjuk (lásd 69. ábra) nagy vonalakban az alkalmazás felépítését rétegekre és eszközökre bontva.

Fő irányelvünk az érthetőség, egyszerű kezelhetőség. Ennek következtében a legtöbb oldalnak egy jól meghatározott szerepe van, hogy ne terheljük túl a kezdő, feltételezhetően a technológiához alig értő felhasználót. Ezen tervezési döntés miatt születtek meg olyan döntések, mint az elkülönült helyszín választó oldal (Select) foglалásokhoz, a kártya alapú kinézet, hogy az egy témához tartozó elemek egy adott kártyán legyenek, valamint az, hogy minden oldalra megpróbálunk működő vissza gombot is helyezni, kivéve, ha egy jól ismert funkcióval állunk szemben (pl. Bejelentkezés, Regisztráció), valamint az olyan oldalak, amiket csak Speciális jogosultságokkal ellátott felhasználók érhetnek el, hiszen róluk feltételezhető, hogy értenek a technikához, amit használnak.

Az applikáció fő elemeinek felépítése a következő (lásd 70. ábra):

A jobb olvashatóság érdekében ez az ábra mellékletként megtalálható lesz a tömörített állományban ClassDiagram.vsdx néven.

A funkciók menetének leírását a következő használati esetdiagramra alapoztam (lásd 71. ábra):



71. ábra: Használati esetdiagram

A jobb olvashatóság érdekében ez az ábra mellékletként megtalálható lesz a tömörített állományban UseCaseDiagramAutoNet.vsdx néven.

Ezekén kívül kialakítottam még néhány felületi tervet is az oldal kinézetének standardizálásához, ezeket a nézetekről szóló fejezetben (Kinézet megtervezése - 3.3.1.1, 72. ábra és 73. ábra).

3.2 Felhasznált technológiák

Mielőtt tüzetesen megvizsgáljuk az alkalmazás rétegeit, beszélnünk kell néhány fontosabb NuGet csomagról, amit használunk:

- EntityFrameworkCore [8] – Az adatbázissal való kapcsolatot ezen keresztül hozzuk létre.
- ASP.NET Core [9] – Alapvető Asp.Net funkciók, fő felhasználás a felhasználó kezelésben.
- Toast értesítések [10] – Felugró ablakcskák, amikkel státusz változásokat jelzünk. Fő csomag: NToastNotify.
- IText7 [11] – PDF generálás.
- EPPlus [12] – Excel táblázat generálása.
- MailKit [13] – Email küldés.
- Hangfire [14] – Emailok ütemezése.
- MSTest [15] és Mock [16] – Tesztelés alapjai.
- Bootstrap [17] – Dinamikus megjelenítéshez nyújt segítséget, alap kinézet stílusa.
- Google Maps [21] – Kisebb és nagyobb térképek megjelenítéséhez.

Az alkalmazást a Visual Studio 2022 Fejlesztési környezettel hoztuk létre, kezdetben .Net 6.0-ra építve, amit később .Net 8.0-vá alakítottunk. Kihasználtuk a Visual Studio nyújtotta lehetőségeket, kifejezetten a Scaffolding [19] funkciót, amivel az egyszerűbb oldalakhoz egy alap vázat generáltunk, valamint amivel behoztuk a felhasználó-kezelés működését és oldalait. Ennek következtében a program ezen része kissé másképp működik, mint a többi, erről részletesen fogunk még beszélni a Felhasználó-kezelés fejezetben (lásd 3.5).

A továbbiakban a rétegek elemeit vesszük sorba, tüzetesebben ismertetjük az alkalmazás részeit.

3.3 Rétegek

Mint már fent említettük, az alkalmazás három fő rétegből áll: Modell, Vezérlés és Nézet (Model-Controller-View architektúra – MVC).

3.3.1 Nézetek

3.3.1.1 Kinézet megtervezése

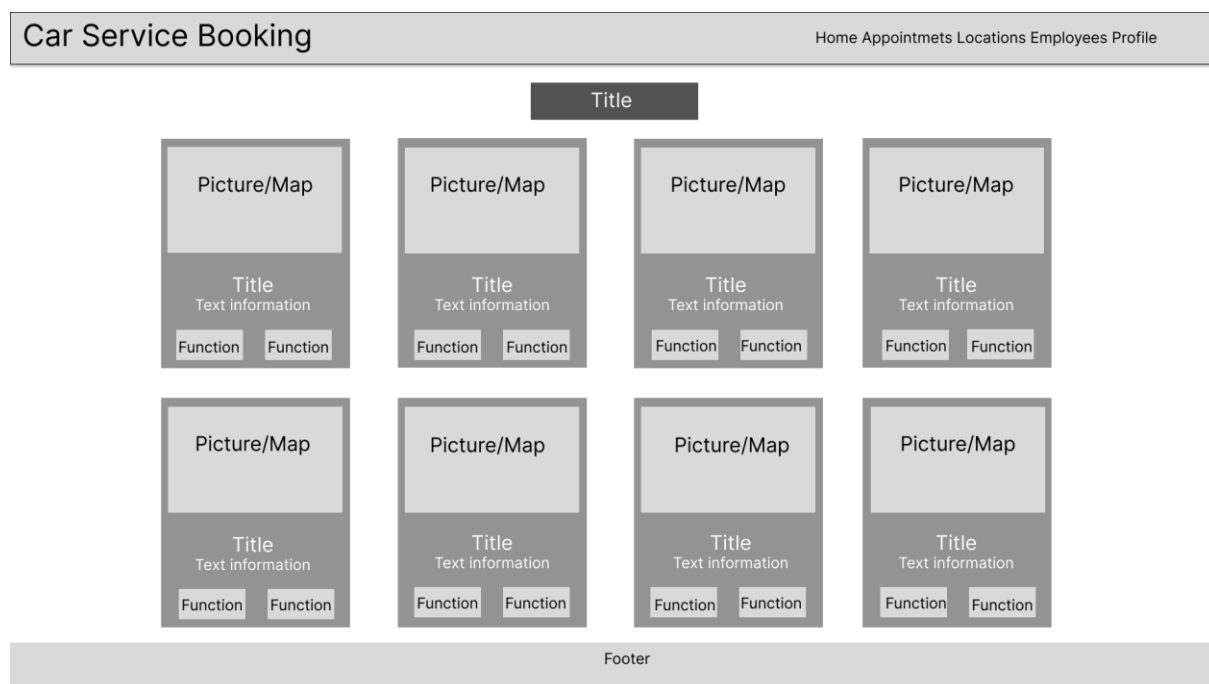
Az applikáció nézeteinek alapja az úgy nevezett „Razor Page” [20], ami kombinálja a HTML és JavaScript nyelveket a C# nyelvvel, így ezekben remekül lehet dinamikusan elemeket létrehozni az adatbázisunk alapján. Kinézet szempontjából fontos szempontok voltak: az érthetőség, az információ szétosztása több oldal között és a könnyű kezelhetőség. Ezek alapján megalkottunk néhány irányelvet, amiket érdemes betartani a további fejlesztés során:

- Mivel egy viszonylag komplex hátteret választottunk a kinézet megtervezése során, a legtöbb szöveg fehér színű, vagy erős narancssárga kell legyen. Előnyben részesítjük a kártyák használatát, mivel ezek jól szemléltetik, hogy melyik elemhez mi tartozik. A két preferált kártya elkészítési módszer a következő:
 - CSS fájlban carsdcontact és ennek változatai – Saját tervezés, jól mutat, ha sok kis kártyát kell megjeleníteni, több szélességgel is megtalálhatóak
 - A bootstrap card bg-dark text-white kártyái letisztultabb megjelenítéshez, a bejelentkező felületek és a legtöbb adatmanipulálási felület így készült
 - A fentihez hozzáadott árnyékosztállyal a cardscontact osztályhoz hasonló, de dinamikusságában másképp viselkedő elemet kapunk.
- Minden oldalnak pontosan egy darab fő funkciót adunk, így nem terheljük túl a felhasználót. Jelen kell legyen legalább egy darab gomb, ami a következő lépéshez juttat, egy, ami visszajuttat, és opcionálisan egyéb funkcionális gomb.
- A fejléc mindig könnyen a felhasználó rendelkezésére kell álljon, hiszen ezzel végezzük a navigáció jelentős részét.

A fentiek alapján minden oldal három fő részre osztható:

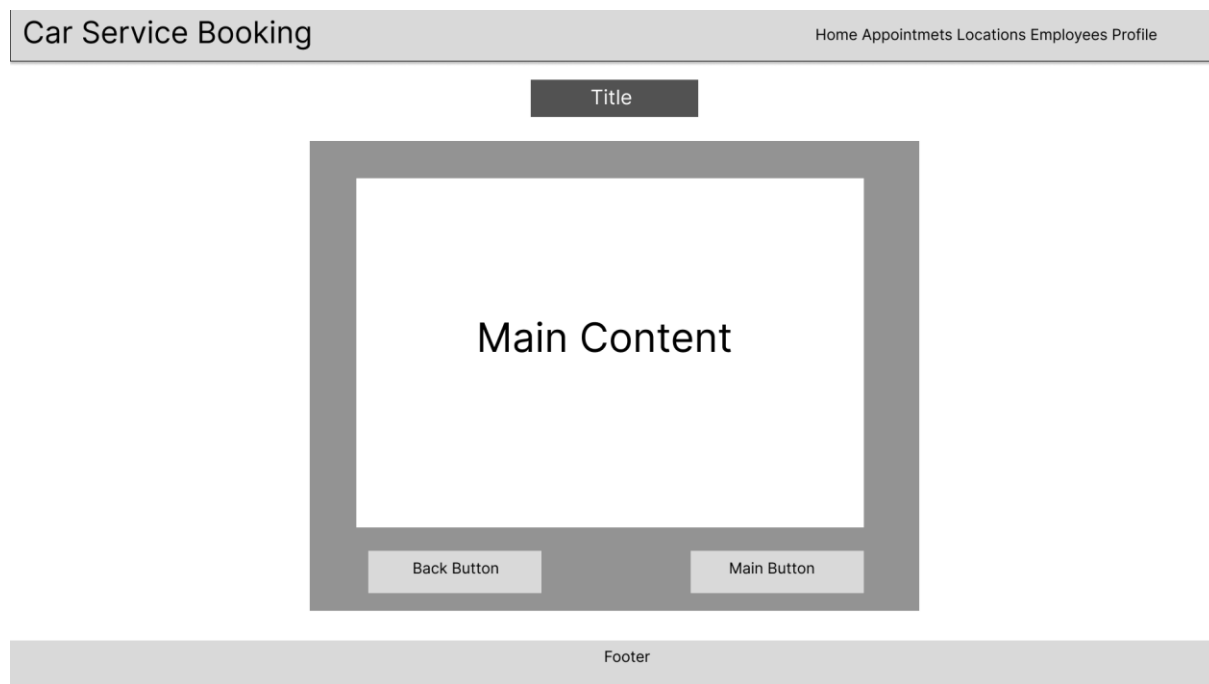
- Fejléc – Itt találhatóak az alap funkciókkal és a fiókkezeléssel kapcsolatos navigációs elemek.
- Tartalom – Itt található a tényleges funkcionalitás, az általános háttéren található kártya vagy kártyák formájában.
- Lábléc – Díszítőelem a szimmetria megőrzése érdekében, kinézetre tükrözi a fejléct.

Ezek az oldalak nagyrészt a következő felületi tervek alapján lettek megtervezve:



72. ábra: Sok kisebb információt bemutató oldal terve

A fenti terv (lásd 72. ábra) olyan oldalakhoz alkalmas, ahol dinamikusan képezünk kis elemeket az adatbázis tartalma alapján. A funkció gombokkal további lehetőségeket érhetünk el az adott elemhez kapcsolódóan. Ilyen például a Helyszín listázó és a Dolgozó listázó oldalak kinézete.



73. ábra: Egy fő funkcióval rendelkező oldal terve

A fenti séma olyan (lásd 73. ábra) esetekben alkalmazható, ahol az oldal egy fontos funkciót lát el. A tagoltsági irányelveink miatt erre sokszor lesz szükség. Erre alapul a foglalási táblázat oldala, az összes adatkészítő és módosító oldal, de részben még a bejelentkezési felület is.

3.3.1.2 Adatmegjelenítés

Az adatmegjelenítés szorosan összefügg a vezérlés réteggel, így az adatok átadásának módjairól bővebben ott fogunk beszélni. Egyelőre elég annyit tudni, hogy az adatmozgatáshoz a „Model” elemeket, a ViewBag-et és a ViewData-t használok, hogy dinamikus módon jeleníthessem meg az adatbázis aktuális állapotát.

Sok oldalhoz tartozik egy @model típus elem, ami a vezérlésben átadott „return view()” függvény paraméterével egyenértékű, elérését a „Model” elemmel teszi lehetővé.

A dinamikusság talán legjobb példája a Foglalásoknál található főoldal (Appointments/Index), ami maximálisan kihasználja a Razor Page dinamikusságát:

- A tábla fejlécében az elkövetkező 5 munkanap nevét kell begeneráljuk, ehhez a JavaScript nyelvben írtunk egy rövid generáló műveletet.
- A tábla tényleges tartalmát egy hosszú C# résszel hozzuk létre, mivel ez teljesen az adatbázis tartalmától függ. Itt rengeteg feltétel alapján minden oszlopot feltöltjük a kívánt tartalommal, ami függ a már eltárolt foglalásoktól és ezek tulajdonosaitól, attól, hogy épp mennyi az idő, a helyszín munkaórától, és attól, hogy be vagyunk-e jelentkezve, vagy hogy van-e Admin jogosultságunk. Itt is figyelembe kell veyük, hogy milyen napok munkanapok, és melyek nem azok.

Részlet a generáló függvényből, ebben a szekcióban generáljuk egy adott elem tartalmát. A végeredmény jobbra található (lásd 74. ábra).

Egy dinamikusság másik érdekes felhasználás a nagy térképnél van, ami a Google Maps API-al kommunikál teljesen csak a JavaScript nyelvre hagyatkozva. Ez a Locations/Bigmap úton érhető el.

3.3.1.3 Adatok visszaküldése.

Rengeteg oldal célja az adatbázis manipulálása. Ezeket a HTML <form> tagjával hozzuk létre, lényegében post metódusokkal. Minden funkcióhoz és minden adattípushoz egyedi oldalakat kell készítenünk, a @modell típusa mindig a visszaküldendő elem típusával egyezik meg.

Ha vannak elemek, amiket automatikusan generálunk, azokat is meg kell említsük ezekben az űrlapokban, de el kell rejtenünk a felhasználó elől, módosító oldalakon érdemes az épp módosítandó elem adatait előre betölteni, ezt az ASP.NET automatikusan el is végzi helyettünk, csak át kell adnunk a vezérlőben az adott elemet. Kivételt képez az előző állítás alól a kép formátumú elem, amit biztonsági okok miatt mégsem adhatunk vissza, valamint a helyszínekhez tartozó dolgozók kiválasztása, akiket a kapcsolta alapúság miatt minden egyes módosítás során újra be kell jelölni. A törlési oldal is lényegében egy űrlap, csak minden elem el van rejtve a felhasználó elől.

3.3.1.4 Részleges nézetek

Az egységes kinézet eléréséhez részleges nézeteket is készítettünk, amik a minden oldalon megjelenő elemeket biztosítják. Legtöbbjük az oldal helyes működését biztosítja, de a Shared mappában van két nézet, ami minden oldalhoz vizuális elemeket is társít:

- `_Layout` – Ez felel a fejléc és a lábléc kinézetéért, valamint a bootstrap stílus és a Toast értesítések beépítéséért.
- `_LoginPartial` – Ez a fejléc kiegészítő része, ebben találjuk a bejelentkezéshez kapcsolódó fejléc elemeket.

| | 9 Apr | 10 Apr | 11 Apr | 12 Apr | 15 Apr |
|----|---------------|--------|----------|--------|--------|
| 8 | Not Available | Free | Free | Free | Free |
| 9 | Not Available | Free | Free | Free | Free |
| 10 | Not Available | Free | Free | Free | Free |
| 11 | Not Available | Free | Free | Free | Free |
| 12 | Not Available | Free | Free | Free | Free |
| 13 | Free | Free | Free | Free | Free |
| 14 | Free | Free | Occupied | Free | Free |
| 15 | Occupied | Free | Free | Free | Free |
| 16 | Occupied | Free | Free | Free | Free |
| 17 | Free | Free | Free | Free | Free |
| 18 | Closed | Closed | Closed | Closed | Closed |
| 19 | Closed | Closed | Closed | Closed | Closed |
| 20 | Closed | Closed | Closed | Closed | Closed |

74. ábra: Foglalási tábla

3.3.1.5 Érdekességek

Különösen figyelniük kell a szövegek elkódolásának módjára, mivel az ASP alapú útvonalkódolás és a JavaScript és HTML alapú kódolás nem mindig egyezik meg. Erre jó példa az aposztróf (') jel, amit explicit át kellett alakítani a helyes működéshez.

Az oldal egy sötétebb bootstrap fájl alapján kapja a színvilágát. Emellett használunk SVG-ket, valamint egy speciális kapcsolót (checkbox), ami a CSS fájlban van definiálva (site.css).

A Google Maps API beépítéséhez egyedi API kulcsra volt szükségünk, ennek későbbi működése erősen fog függeni a Google fejlesztéseitől.

3.3.2 Modell

A modell réteg tartalmazza az összes olyan SQL tábla definícióját, amikben a működéshez szükséges adatokat tároljuk. Ezeknek három típusa van:

- Fő adattároló – Ezek valós adatok absztrakcióját tartalmazzák. Ide helyezhetőek az Appointments, Employees és Locations táblák
- Hiba adat – Ebben biztosítjuk a hibát jelző oldalak megjelenítéséhez szükséges adatokat. Ide tartozik az ErrorViewModel tábla.
- Kisegítő táblák – Ezek olyan táblák, amikben nem egyszerű egyedeket, hanem kapcsolatokat tárolunk. Ide tartozik a LocationEmployee tábla.

3.3.2.1 Fő adattárolók

A fő adattárolók a program lelkét képezik. Ezek a táblák biztosítják a mentés lehetőségét a program további részeinek, nagyrészt egyszerű adatelemekből állnak (kivétel a LocationEmployee adatelem).

Minden szükséges ellenőrzés a Vezérlésben biztosítva van, így ide biztos csak helyes adatok kerülhetnek.

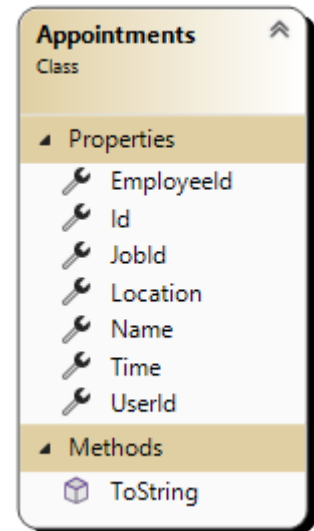
A fontos adatokat a **System.ComponentModel.DataAnnotations** osztály segítségével jelöljük. Szerkezetük a következő:

Appointments

Egy foglalás adatai (lásd 75. ábra). A legtöbb elem egyszerű adatokat ad meg, vagy kapcsolatot teremt más osztályokkal (Location, EmployeeId). Kulcs: Id. Kiírás egyszerűsítése érdekében jelen van egy általános ToString metódus is, ami a kiírandó szöveggel (string) tér vissza

Érdekesebb elemek:

- UserId – a felhasználó email címe, aki készítette a foglalást. Ezt használják fel az instant emailt küldő függvények.
- EmployeeId – ki dolgozik ezen a munkán
- JobId – Ez a beütemezett emailhez tartozó azonosító



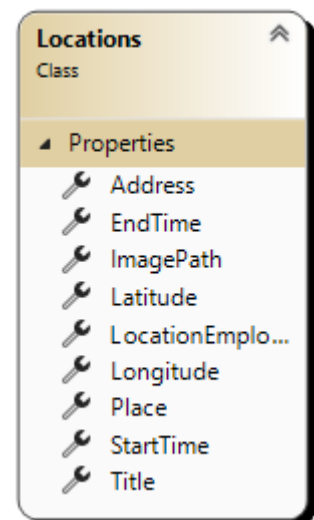
75. ábra: Appointments modell osztály

Locations

Egy helyszín (műhely) adatai (lásd 76. ábra):

Érdekesebb elemek:

- Place – Helyszín neve, ez a kulcs.
- Latitude és Longitude – Ezen adatok alapján kerül fel a helyszín a térképre
- ImagePath – Az előre megadott reklámképek egyikére mutató útvonal. Ezek a **wwwroot/css/Media** mappában található előre megadott képek
- LocationEmployees – A helyszínhez tartozó dolgozók leírására szolgál, erről később bővebben beszélünk.



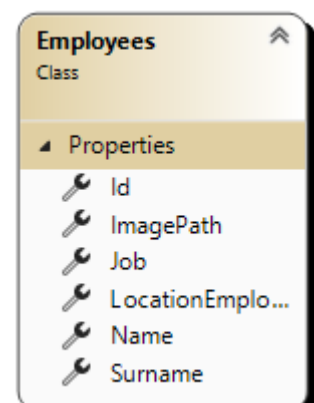
76. ábra: Locations modell osztály

Employees

Egy dolgozó adatai (lásd 77. ábra):

Érdekesebb elemek:

- Id – Egyedi szám azonosító, kulcs.
- ImagePath – Útvonal az adott dolgozó feltöltött képéhez, ami készítéskor töltünk fel. Ez elmentve tárolódik a **wwwroot/css/Media** mappában.

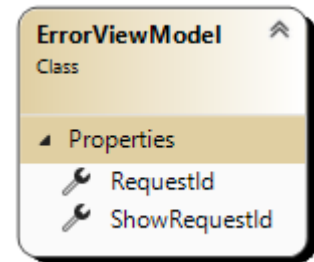


77. ábra: Employees modell osztály

- LocationEmployees – Itt tárolódik, hogy az adott személy mely helyeken dolgozik.

3.3.2.2 Hiba adat

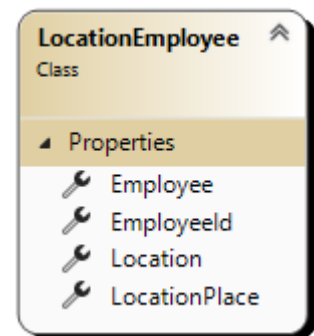
Ez az osztály (lásd 78. ábra) biztosítja a hiba oldalak behozását, a vezérlés hibáinak dobásában segít, az Error nézet használja.



88. ábra: ErrorViewModel osztály

3.3.2.3 Kisegítő táblák

Jelenleg a rendszer egy darab kapcsolótáblát alkalmaz: a LocationEmployee táblát (lásd 79. ábra). Ez a tábla sok-sok kapcsolatokat tesz lehetővé helyszínek és dolgozók között. Első sorban az időpontfoglalásban használjuk, ahol a felhasználónak lehetősége van kiválasztani, hogy melyik dolgozóval szeretné elvégeztetni a kívánt munkálatokat.



79. ábra: LocationEmployee kapcsolati model osztály

Két fő elemtípusa van:

- Identifier properties - Olyan elemek, amik egyértelműen meghatározzák az adott helyszínt és dolgozót. Ezek a LocationPlace és az EmployeeId elemek. Ezek a könnyebb azonosítás céljával készültek, így nem kell minden egyes ellenőrzésnél az eredeti osztályba belelépni, elég csak ezeket az adattagokat összevetni a meglévő példányok kulcsaival.
- Navigation properties – Olyan elemek, amik a megjelölt elemek elérését biztosítják. Ezek a Location és Employee elemek.

Helyszín létrehozásakor opcionális ezt megadni, ilyenkor a legördülő menüben kiírjuk, hogy a helyszín nem használja ezt a funkciót, egy -1 Id kerül majd a foglalás EmployeeId mezőjébe. A sikeres paraméterezést az ApplicationDbContext osztály biztosítja, erről a következő fejezetben fogunk beszélni.

3.3.3 Adat (Data) és Adatvándorlás (Migrations)– Kitérő

Bár ezek nem részei az MVC paradigmában felsorolt három fő rétegnek, a program működéséhez mégis elengedhetetlen elemek a Data és Migrations mappák. Ezek kiegészítő osztályok a modell helyes működéséhez. Mivel „Code First” módon építjük az adatbázist, a felépítés módját is szükséges definiálnunk.

3.3.3.1 Adatvándorlás

Ezek a fájlok egymásra épülnek, azt mutatják meg, hogyan építhetjük fel a megfelelő adatbázist. Első sorban azért tartjuk meg, mivel, ha a modell réteg fejlesztése során komolyabb hibát vétünk, vissza tudunk építeni egy olyan adatbázist, amivel még helyesen működtek a funkciók. Legnagyobb eleme az első (Initial), ebben található az első változata az adatbázisnak, az ezután következők csak inkrementálisan alakítják az adatbázist.

Két fő függvényük van:

- Up – Készítés paraméterei
- Down – Lebontás paraméterei

Ezek szorosan összefüggenek, minden felépített elemhez létre kell hoznunk a szükséges lebontási módszereket is, hogy visszakerüljünk az ezelőtti Adatvándorlás helyzetébe.

Generálás

Az ASP.NET rendszer egyszerű megoldást kínál az adatvándorlások létrehozására a **Microsoft.EntityFrameworkCore** csomagon keresztül. A csomag telepítése után két parancsot kell a csomag menedzselő konzolba (Package Manager Console) írunk:

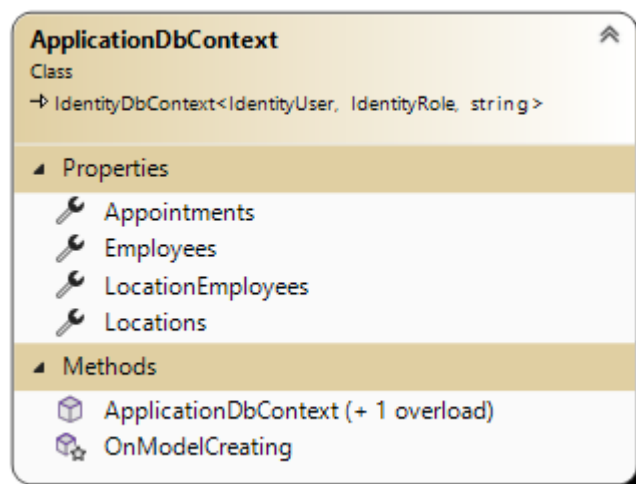
- Add-Migration <név> - Ezzel a rendszer automatikusan létrehozza az új adatvándorlási fájlt
 - A Remove-Migration paranccsal ezt visszacsinálhatjuk
- Update-Database – Ezzel frissítjük a rendszerhez tartozó adatbázist, aminek elérési útvonalát az appsettings.json fájlban definiáltuk.

A fenti string a localhost szerverhez van, amiben a fejlesztés és futtatás során teljesen megbízhatunk, így minden ellenőrzést automatikusan átlépünk:

Trusted_Connection=true;TrustServerCertificate=true

3.3.3.2. Adat

Ebben a részben hozzuk létre az ApplicationDbContext osztályt (lásd 80. ábra), ami a Microsoft.EntityFrameworkCore csomag segítségével egy átjárót nyújt az adatbázis és a Modell részleg között. Ez tartalmazza a tényleges adatokat, létrehozza a modell elemeit, valamint konfigurálja a speciális osztályokat, jelen esetben a LocationEmployee osztályt.



80. ábra: Az ApplicationDbContext adat osztály

Elemként tartalmazza az összes elérhető osztályt, így a vezérlésben az adatok elérését ennek az osztálynak egy példányán fogjuk végezni.

3.3.4 Vezérlés (Controller)

A Controller mappában találjuk a vezérlési fájlokat, amik összeköttetést biztosítanak a Megjelenítés és az Adat rétegek között. Minden főbb funkcióhoz külön Vezérlés fájl tartozik (pl. AppointmentsController – Foglalási oldalakért felel).

A függvények nevei, meghatározzák a hatókörüket, szabály az, hogy az oldalakkal foglalkozó függvények nevei megegyeznek az oldal View fájljának nevével, annotációkkal jelezzük, hogy ez egy Get vagy egy Post metódus lesz-e.

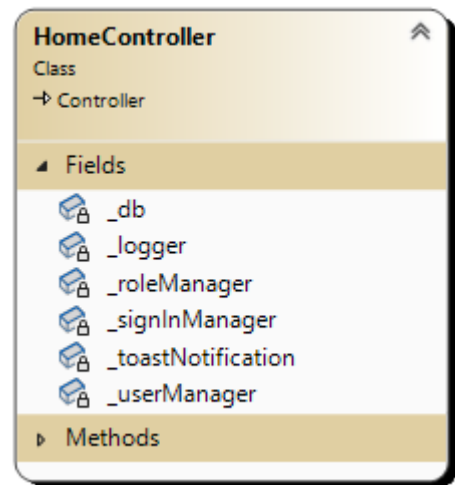
4 fő kontroller osztályunk van:

- HomeController – Főoldal és kisebb, egy oldalas funkció, applikáció-specifikus Adatbázis módosítása nélkül (lásd 83. ábra).
- AppointmentsController – A foglalási felületek működtetése és az ezekhez kapcsolódó funkciók biztosítása (Excel, Pdf generálás).
- LocationsController – A helyszín felületek kezelése.
- EmployeesController – A dolgozók felületeinek kezelése.

3.3.4.1. Adatelemek

A fájlok adatalemeinek típusai a következők:

- ApplicationDbContext – Fő elem, az adatbázis elérést biztosítja.
- ILogger<AppointmentsController> - Log üzenetek kezeléséhez
- INotyfService – Toast értesítések kezelése
- SntpSettings és IConfiguration –Sntp beállítások eléréséhez, ezek az appsettings.json fájlban vannak definiálva, a Connection String alatt.



81. ábra: A HomeController osztály adatalemei

- IWebHostEnvironment – A futtató rendszerről kapott információ, képek mentési helyének elérésében segít a Dolgozóknál
- UserManager<IdentityUser>, RoleManager<IdentityRole> és SignInManager<IdentityUser> - Felhasználó kezeléshez szükséges elemek, jogosultságok kiosztása, felhasználó előkeresése, bejelentkeztetés.

Jó példa ezen elemekre a HomeController osztály (lásd 81. ábra)

3.3.4.2. Funkciók

Ezen fájlok nagyrészt egyformák, funkcióik a következők:

- Oldalak helyes betöltése, adatok sikeres átadása – Get
 - Két fő részből állnak:
 - Megjelenítendő adatok összeállítása (ha szükséges)
 - Adatok átvitele – Return View(adatok)
 - Az adatátvitel történhet több módon:
 - Speciális osztályban az adat átvitele, ha nem várunk visszatérést (Post metódust), itt szükséges a jó paraméterezés és a speciális elem
 - Adatátvitel ViewBag-en keresztül, mint alobjektum.
 - Elérés: ViewBag.Adatelem
 - Adatátvitel ViewData-n keresztül, mint Map elem.
 - Elérés: ViewData[“Adatelem”]
 - A fejlesztés folyamán alakultak a preferenciák, a végső stádiumban a ViewBag volt előnyben részesítve, mivel kevesebb

probléma adódott vele az adatok elérésénél, mint a ViewData használatánál. Ez főleg személyes kényelmi dolog, személy szerint sokkal egyszerűbben tudtam a ViewBag-el dolgozni.

- Ezek az [HttpGet] annotációval vannak ellátva, IActionResult típussal térnek vissza, az adott oldal betöltésével végződnek.
- Oldalról érkező információk feldolgozása, ellenőrzése és átadása – Post (82)
 - Ezeket a view egy <form> elem post metódusaként adja át.
 - Ezek a függvények három fő részből állnak:
 - Adatok ellenőrzése – Általában egyszerű, statikus ellenőrzések
 - Pl. Hosszúsági és szélességi fokok értékei helyesek-e?
 - Adatbázis frissítése
 - Az ApplicationDbContext példányon keresztül hozzáadás, törlés.
 - Új oldal megjelenítése – Return View(adatok) vagy RedirectToAction(oldal).
 - Ezekhez sokszor tartozik egy Email küldő elem is, amikhez a Mailkit és Hangfire csomagokat használjuk. Ezek általában a SendEmailAsync (ha az smtp beállítások alapján küldünk, Mailkit instant küldés), vagy a SendEmailAsyncFixed (ha a Hangfire késleltetett üzeneteivel írjuk, beégetett szerverről, mivel a Hangfire csak így volt működtethető). Ezek a funkciók lehetnek az EmailMethods statikus segédosztályban, vagy egyenesen a vezérlő osztályba beépítve.
 - Annotációk: [HttpPost] és [ValidateAntiForgeryToken] a biztonságos adatátadás érdekében, IActionResult-al térnek vissza, az adatmódosítás után a következő oldalra való átirányítással végződnek.
- Funkcionális betöltő gombok – Általában a NuGet csomagok által biztosított funkciókhoz készített függvények, az adott csomag egyedi függvényeivel dolgozva
 - Pdf generálás – GeneratePdf, GeneratePersonPdf
 - Ezek is IActionResult-tal térnek vissza, a pdf fájl létrehozásával járnak.
 - Excel generálás – GenerateExcel, GeneratePersonExcel (83)
 - Ezek is IActionResult-tal térnek vissza, a Excel fájl létrehozásával járnak.

A sima generáló a teljes adatbázissal dolgozik, a „Person” verziók csak az adott felhasználó adatait veszik figyelembe.

3.3.4.3. Érdességek

A program fejlesztése során sok különböző technológia működését össze kellett hangolni, így vannak érdekesebb megoldások, amiket érdemes kiemelni.

Figyelnünk kell, hogy a fiók-specifikus műveletek gombjai csak akkor legyenek elérhetők, ha ténylegesen be van valaki jelentkezve. A User.Identity elemen keresztül érjük el az épp bejelentkezett személyt, de ez lehet Null is, feltéve, hogy nincs senki belépve, így ezek a funkciók csak akkor szabad elérhetőek legyenek, ha valaki ténylegesen be van jelentkezve.

Vannak a fejlesztés során végül fel nem használt, vagy épp fejlesztés alatt álló elemek is a programban, amikkel szépíthetjük a működését, valamint kikommentelt belső tesztelő elemek is, amik felhasználhatóak további fejlesztés során.

Az email küldést a Google Gmail rendszerében létrehozott email címmel bonyolítjuk le, a Gmail fejlesztésének menete erősen befolyásolhatja a helyes működést a jövőben (vagy, hogy egyáltalán működni fognak-e ezek a funkciók).

3.4 A főprogram

A Program.cs fájlban találjuk az applikáció alap konfigurációjának eléréséhez szükséges műveleteket. Ebben az osztályban két fontos adatelemet találunk:

- Builder – Ez felel az alkalmazás felkonfigurálásához szükséges adatok befogadásáért. Itt adhatjuk meg az alapvető kapcsolatot az adatbázissal (Ez a DefaultConnection, amit az appsettings.json fájlban definiáltunk), valamint itt készítjük elő a felhasznált technológiákat is (pl Toast értesítések, Hangfire)
- App – Ez felel a tényleges rendszerek elindításáért, és ezek további állításáért. Ezen keresztül kapcsoljuk be a már működő rendszerek funkcióit is, például a Hangfire megfigyelő oldalát (app.UseHangfireDashboard).

Az osztály utolsó parancsa mindig az app.Run() kell legyen.

3.5 Felhasználó kezelés

A felhasználó kezelés a Visual Studio és a .NET 8.0 beépített Identity „scaffold”-jára alapul, kiegészítve az emailek elküldésének képességével, amit a MailKit segítségével oldunk meg.

3.5.1 Felépítés

A scaffold egy kissé kellemetlen velejárója, hogy a funkcionalitás a Razor-Pagek C# blokkjában találhatóak, nem pedig egy külön kontrollerben, így ezekben vannak megjelenítve a működési logikához komolyabban kapcsolódó függvények is. Két fő függvényünk van:

- OnGet(Async) – Ez határozza meg az oldal adatbetöltését és működését betöltés során.
- OnPost(Async) – Ez határozza meg a program működését, ha egy névtelen post metódusú kérdőív (form) beküldésre került.

A fent említett metódusok mellett hozzáadtuk még az emailek küldéséhez szükséges metódust is ezekhez az osztályokhoz, amit SendEmailAsync-nek nevezünk. Paraméterei a következők:

- To – Kinek küldjük (email cím)
- Subject – Az Email címe, lényegi összefoglaló
- Body – Az Email tartalma

Ezek beégetett információként is megjelenhetnek, de az smtp beállítások alapján is megadhatóak. Fontos, hogy a Hangfire csak a beégetett verzióval működik, mivel nem tud az smtp információk osztályából példányt készíteni küldéskor.

3.5.2 Szerepkörök

Minden felhasználóhoz tartoznak jogosultságok, amik alapján elérhetővé válnak számára további funkciók. Az ellenőrzés nagyon egyszerű, az épp bejelentkezett felhasználó jogosultsága alapján jelennek meg a funkciók eléréséhez szükséges gombok.

A szerepköröket feljebb már említettük (lásd 2.4 – Felhasználó típusok), itt kiemelendő az Admin szerepkör, ami alapból egy admin@admin.com felhasználóhoz kerül. Ezen fiókot létrehozva ez automatikusan megkapja az Adminisztrátor jogosultságot, bár az adatbázis állításával előbb aktiválni kell majd, hiszen ez nem egy valódi email cím, ennek menetét az Adatbázisról szóló részlegben tüzetesebben leírjuk. Innen ez befolyásolhatja a többi regisztrált felhasználó jogosultságait az All Users menüponttal.

3.5.3 Védelem

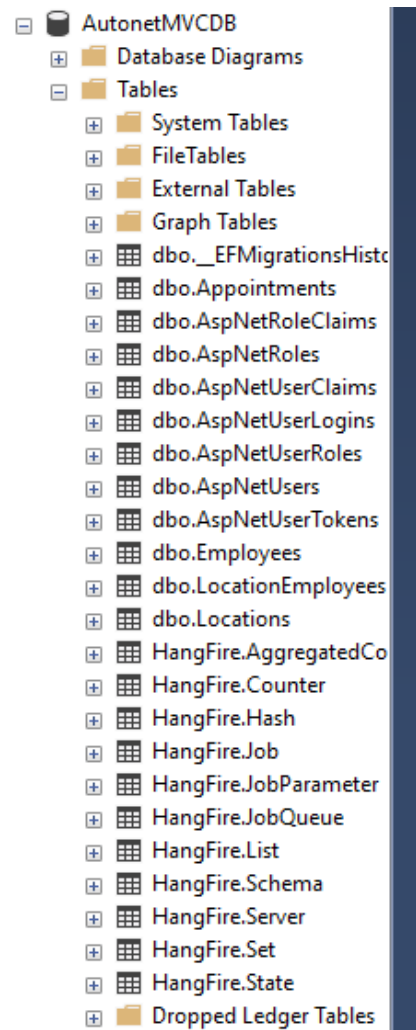
Minden felhasználónak meg kell erősítenie email címét, mielőtt hozzáférhetne az oldalhoz. Ha valamilyen hiba történne, biztosítanunk kell, hogy az adatokat könnyen vissza lehessen szerezni az elfelejtett jelszó funkcióval, valamint az igazoló email újra küldését is meg kell oldanunk szükség esetén. Ezek a funkciók jelenleg elérhetőek a belépés oldalon.

A fiók teljes átalakításának és törlésének lehetőségét is mindenképp meg kell őriznünk, nem akarunk fölöslegesen adatokat tárolni olyan felhasználókról, akik kényelmetlenül érzik magukat ezzel kapcsolatban.

3.6 Az adatbázis

A rendszer háttérben egy Microsoft SQL adatbázis áll, amit a localhost címen működtetünk, és AutonetMVCDB néven fut (lásd 82. ábra). Ennek tábláiban tartjuk a program működéséhez elengedhetetlen adatokat, ezek a táblák négy csoportba sorolhatóak:

- Modell táblák – a dbo osztályba tartoznak, ezek felelnek az applikációba betáplált adatok megőrzéséért és visszaadásáért. A modell minden eleméhez tartozik egy ilyen tábla, kivéve az ErrorViewModellt.
- Felhasználó kezelő táblák – a dbo osztály ASP.NET szóval kezdődő elemei, ezek felelnek a felhasználók adataiért (AspNetUser) és jogosultságaiért (AspNetRole). Ezek a scaffolding során keletkeztek. Meg kell jegyeznünk, hogy a manuális email igazolást a dbo.AspNetUsers EmailConfirmed oszlopában található elem átírásával érhetjük el (a 0 nincs igazolva, az 1 igazolva van.)
- HangFire táblák – ezekben a táblákban tárolódnak a beütemezett munkák. Főleg az emailek ütemezett küldéséhez használjuk fel.



82. ábra: Az adatbázis felépítése

3.7 Tesztelés

Az applikáció tesztelését a megoldásban (solution) található MSTest projekttel végezzük. A tesztelésben főleg a vezérlési (controller) réteg fontosabb függvényeit ellenőrizzük.

A tesztelésben felhasználunk Mock objektumokat, hogy leegyszerűsítsük a tesztek létrehozását, minden egyes vezérlési egység tesztelését pedig külön osztály fájlban bonyolítjuk le, valamint egy közös egységben a modell elemeit is teszteljük, így összesen öt darab osztályra lesz szükségünk, hiszen első sorban a vezérlés elemeit szeretnénk tesztelni.

Minden fájl tartalmaz egy előkészítő függvényt is, amiben létrehozuk, és feltöltjük a felhasznált Mock objektumokat, így a tesztfüggvényekben már csak a műveletek végrehajtását és az eredmények tesztelését kell lebonyolítanunk.

Általánosságban a tesztelésünk az Elrendezés-Tevékenység-Ellenőrzés (Arrange-Act-Assert) szerkezetre alapul, így minden függvényünk érthető módon fog viselkedni, az összes fontos tevékenységet végre fogják hajtani bármilyen ellenőrzés előtt.

3.7.1 Szerkezet – Unit tesztelés

Minden tesztelési fájl három fontos részből áll:

- Adatelemek – Ezek lehetnek a vezérlés létrehozásához szükséges báb (Mock) objektumok, valamint a tényleges vezérlési egységet alakító objektum található.
- Konfiguráció – Ezt egy Setup nevű függvényben tesszük meg. Itt hozzuk létre az alapvető adatokat, amikkel később dolgozni fogunk, amiket az adatbázisunkat jelképező bábba fogunk építeni, valamint a vezérlési egységet, és az ehhez elengedhetetlen elemeket is itt alkotjuk meg. Itt végezzük el a tesztelésünk elrendezési (Arrange) lépését.

Fontos megjegyezni, hogy a futáshoz felkonfigurált útvonalat kissé alakítanunk kell a helyes működéshez a konfiguráló függvényekben.

- Teszt funkciók – Ezekben folytatjuk le a tényleges tesztelés történéseit. A vezérlési fájlok minden komolyabb metódusára megpróbálunk valamilyen egyszerű ellenőrzést írni, ami biztosít minket arról, hogy a metódusok megfelelően működnek.

3.7.2 Manuális tesztelés

Az applikáció a fejlesztés során tüzetes tesztelésen esett át, minden funkció újra és újra ki lett próbálva különböző adatokkal, hogy biztosak lehessünk a jó működésben. Emiatt sok kis apróságot ki tudtunk küszöbölni, jó példa erre a HTML és JavaScript közötti link kódolási különbségek felfedezése (lásd 3.3.1.5 – Érdekességek). A használati esetdiagram (lásd 71. ábra) minden egyes ágát többször is végig jártam, és ha felmerült valamilyen inkonzisztencia, javítottam.

A manuális tesztelés menete a következő User Story táblázatok alapján volt lebonyolítva (lásd 4.-19. táblázatok):

| | | |
|---|-----------|---|
| | AS A | Vendég felhasználó |
| | I WANT TO | Bejelentkezni |
| 1 | GIVEN | A weboldalon vagyok |
| | WHEN | Megnyomom a bejelentkezés gombot |
| | THEN | A bejelentkezési felületre kerülök |
| 2 | GIVEN | A bejelentkezési felületen vagyok |
| | WHEN | Jó adminisztrátor adatokat adok meg |
| | THEN | Adminisztrátorként bejelentkezek, főoldalra kerülök |
| 3 | GIVEN | A bejelentkezési felületen vagyok |
| | WHEN | Jó felhasználó adatokat adok meg |
| | THEN | Bejelentkezek, főoldalra kerülök |
| 4 | GIVEN | A bejelentkezési felületen vagyok |
| | WHEN | Rossz adatokat adok meg |
| | THEN | A hiba jelezve, ismét beírhatom az adataim |

4. táblázat: Teszteléshez használt User Story táblázat, bejelentkezés

| | | |
|---|-----------|--|
| | AS A | Bármilyen felhasználó |
| | I WANT TO | Időpontokat megtekinteni |
| 1 | GIVEN | A weboldalon vagyok |
| | WHEN | Megnyomom az időpontok gombot |
| | THEN | A helyszínválasztó felületre kerülök |
| 2 | GIVEN | A helyszínválasztó felületen vagyok |
| | WHEN | Kiválasztottam egy helyszínt |
| | THEN | A helyszín időpont-táblázatát megtekinthetem |

5. táblázat: Teszteléshez használt User Story táblázat, időpontok megtekintése

| | | |
|---|-----------|--|
| | AS A | Bármilyen felhasználó |
| | I WANT TO | Helyszíneket megtekinteni |
| 1 | GIVEN | A weboldalon vagyok |
| | WHEN | Megnyomom az helyszínek gombot |
| | THEN | A helyszín-felsoroló felületre kerülök |
| 2 | GIVEN | A helyszín-felsoroló felületen vagyok |
| | WHEN | Kiválasztottam egy helyszín részleteinek gombját |
| | THEN | A helyszín részleteit megtekinthetem |

6. táblázat: Teszteléshez használt User Story táblázat, helyszínek megtekintése

| | | |
|---|-----------|---|
| | AS A | Vendég felhasználó |
| | I WANT TO | Regisztrálni |
| 1 | GIVEN | A weboldalon vagyok |
| | WHEN | Megnyomom a regisztráció gombot |
| | THEN | A regisztrációs felületre kerülök |
| 2 | GIVEN | A regisztrációs felületen vagyok |
| | WHEN | Jó adatokat adok meg |
| | THEN | Bejelentkezek az új fiókba, főoldalra kerülök |
| 3 | GIVEN | A regisztrációs felületen vagyok |
| | WHEN | Rossz adatokat adok meg |
| | THEN | A hiba jelezve, ismét beírhatom az adataim |

7. táblázat: Teszteléshez használt User Story táblázat, regisztráció

| | | |
|---|-----------|---------------------------------------|
| | AS A | Vendég felhasználó |
| | I WANT TO | Dolgozókat megtekinteni |
| 1 | GIVEN | A weboldalon vagyok |
| | WHEN | Megnyomom az dolgozók gombot |
| | THEN | A dolgozó-felsoroló felületre kerülök |

8. táblázat: Teszteléshez használt User Story táblázat, dolgozók megtekintése

| | | |
|---|-----------|---------------------------------|
| | AS A | Bármilyen felhasználó |
| | I WANT TO | Vissza akarok térni a főoldalra |
| 1 | GIVEN | A weboldalon vagyok |
| | WHEN | Megnyomom az főoldal gombot |
| | THEN | A főoldalra felületre kerülök |

9. táblázat: Teszteléshez használt User Story táblázat, vissza a főoldalra

| | | |
|---|-----------|--|
| | AS A | Felhasználó |
| | I WANT TO | Kijelentkezni |
| 1 | GIVEN | A weboldalon vagyok |
| | WHEN | Megnyomom a kijelentkezés gombot |
| | THEN | Kijelentkeztem, vendég pozícióba kerülök |

10. táblázat: Teszteléshez használt User Story táblázat, kijelentkezés

| | | |
|---|-----------|--|
| | AS A | Felhasználó |
| | I WANT TO | Időpontot foglalni |
| 1 | GIVEN | A weboldalon vagyok |
| | WHEN | Megnyomom az időpontok gombot |
| | THEN | A helyszínválasztó felületre kerülök |
| 2 | GIVEN | A helyszínválasztó felületen vagyok |
| | WHEN | Kiválasztottam egy helyszínt |
| | THEN | A helyszín időpont-táblázatát megtekinthetem |
| 3 | GIVEN | A helyszín időpont-táblázatánál vagyok |
| | WHEN | Megnyomok egy üres időpontot |
| | THEN | Részleteket megadó felületre kerülök az adott időponthoz |

| | | |
|---|-------|---|
| 4 | GIVEN | A részleteket megadó felületen vagyok az adott időponthoz |
| | WHEN | Jó adatokat adok meg |
| | THEN | Lefoglalom az időpontot, visszakerülök az időpont-táblázathoz |
| 5 | GIVEN | A részleteket megadó felületen vagyok az adott időponthoz |
| | WHEN | Rossz adatokat adok meg |
| | THEN | Hibát kapok, helyben maradok |
| 6 | GIVEN | A helyszín időpont-táblázatánál vagyok |
| | WHEN | Megnyomok egy más által foglalt/nem lehetséges időpontot |
| | THEN | Semmi nem történik |

11. táblázat: Teszteléshez használt User Story táblázat, időpont foglalása

| | | |
|---|-----------|---|
| | AS A | Felhasználó |
| | I WANT TO | Saját részleteimet szeretném megtekinteni |
| 1 | GIVEN | A weboldalon vagyok |
| | WHEN | Megnyomom a felhasználónevem |
| | THEN | A saját fiókom részleteihez kerülök |

12. táblázat: Teszteléshez használt User Story táblázat, részletek

| | | |
|---|-----------|--|
| | AS A | Felhasználó |
| | I WANT TO | Saját időpontot szeretnék törölni |
| 1 | GIVEN | A weboldalon vagyok |
| | WHEN | Megnyomom az időpontok gombot |
| | THEN | A helyszínválasztó felületre kerülök |
| 2 | GIVEN | A helyszínválasztó felületen vagyok |
| | WHEN | Kiválasztottam egy helyszínt |
| | THEN | A helyszín időpont-táblázatát megtekinthetem |
| 3 | GIVEN | A helyszín időpont-táblázatánál vagyok |
| | WHEN | Megnyomok egy saját foglalt időpontot |
| | THEN | Törölöm a foglalást |

13. táblázat: Teszteléshez használt User Story táblázat, időpont törlése

| | | |
|---|-----------|---|
| | AS A | Adminisztrátor |
| | I WANT TO | Helyszínt szeretnék törölni a helyszínek listájából |
| 1 | GIVEN | A weboldalon vagyok |
| | WHEN | Megnyomom a helyszínek gombot |
| | THEN | A helyszín-felsoroló felületre kerülök |
| 2 | GIVEN | A helyszín-felsoroló felületen vagyok |
| | WHEN | Kiválasztottam egy helyszín törlés gombját |
| | THEN | Törölöm a helyszínt |

14. táblázat: Teszteléshez használt User Story táblázat, helyszín törlése

| | | |
|---|-----------|---|
| | AS A | Adminisztrátor |
| | I WANT TO | Helyszínt szeretnék hozzáadni a helyszínek listájához |
| 1 | GIVEN | A weboldalon vagyok |
| | WHEN | Megnyomom a helyszínek gombot |
| | THEN | A helyszín-felsoroló felületre kerülök |
| 2 | GIVEN | A helyszín-felsoroló felületen vagyok |

| | | |
|---|-------|--------------------------------------|
| | WHEN | Kiválasztottam az új helyszín gombot |
| | THEN | Helyszín-készítő felületre kerülök |
| 3 | GIVEN | A Helyszín-készítő felületen vagyok |
| | WHEN | Megadtam a kívánt adatokat |
| | THEN | Új helyszínt hoztam létre |

15. táblázat: Teszteléshez használt User Story táblázat, helyszín hozzáadása

| | | |
|---|-----------|--|
| | AS A | Adminisztrátor |
| | I WANT TO | Helyszínt szeretnék módosítani a helyszínek listájában |
| 1 | GIVEN | A weboldalon vagyok |
| | WHEN | Megnyomom a helyszínek gombot |
| | THEN | A helyszín-felsoroló felületre kerülök |
| 2 | GIVEN | A helyszín-felsoroló felületen vagyok |
| | WHEN | Kiválasztottam egy helyszín módosítás gombját |
| | THEN | Helyszín-módosító felületre kerülök |
| 3 | GIVEN | A Helyszín- módosító felületen vagyok |
| | WHEN | Megadtam a kívánt adatokat |
| | THEN | Módosítottam a helyszín adatait |

16. táblázat: Teszteléshez használt User Story táblázat, helyszín módosítása

| | | |
|---|-----------|--|
| | AS A | Adminisztrátor |
| | I WANT TO | Dolgozót szeretnék törölni a dolgozók listájából |
| 1 | GIVEN | A weboldalon vagyok |
| | WHEN | Megnyomom a dolgozók gombot |
| | THEN | A dolgozó-felsoroló felületre kerülök |
| 2 | GIVEN | A dolgozó-felsoroló felületen vagyok |
| | WHEN | Kiválasztottam egy dolgozó törlés gombját |
| | THEN | Törölöm a dolgozót |

17. táblázat: Teszteléshez használt User Story táblázat, dolgozó törlése

| | | |
|---|-----------|---|
| | AS A | Adminisztrátor |
| | I WANT TO | Dolgozó szeretnék hozzáadni a dolgozók listájához |
| 1 | GIVEN | A weboldalon vagyok |
| | WHEN | Megnyomom a dolgozók gombot |
| | THEN | A dolgozó-felsoroló felületre kerülök |
| 2 | GIVEN | A dolgozó-felsoroló felületen vagyok |
| | WHEN | Kiválasztottam az új dolgozó gombot |
| | THEN | Dolgozó-készítő felületre kerülök |
| 3 | GIVEN | A dolgozó-készítő felületen vagyok |
| | WHEN | Megadtam a kívánt adatokat |
| | THEN | Új dolgozót adtam a rendszerhez |

18. táblázat: Teszteléshez használt User Story táblázat, dolgozó hozzáadása

| | | |
|---|-----------|---|
| | AS A | Adminisztrátor |
| | I WANT TO | Dolgozót szeretnék módosítani a dolgozók listájában |
| 1 | GIVEN | A weboldalon vagyok |
| | WHEN | Megnyomom a dolgozók gombot |
| | THEN | A dolgozó-felsoroló felületre kerülök |
| 2 | GIVEN | A dolgozó-felsoroló felületen vagyok |
| | WHEN | Kiválasztottam egy dolgozó módosítás gombját |

| | | |
|---|-------|-------------------------------------|
| | THEN | Dolgozó-módosító felületre kerülök |
| 3 | GIVEN | A dolgozó-módosító felületen vagyok |
| | WHEN | Megadtam a kívánt adatokat |
| | THEN | Módosítottam a dolgozó adatait |

19. táblázat: Teszteléshez használt User Story táblázat, dolgozó módosítása

3.7.3 Érdekességek

Fontos megjegyezni, hogy a .NET nem minden osztályát tudjuk egy az egyben Mock objektummal reprezentálni, emiatt szükségünk lehet olyan segédosztályok elkészítésére, amik ezeket a szükséges osztályokat helyettesítik. Erre jó példa a bejelentkezést helyettesítő FakeSignInManager osztály, ami a bejelentkezési próbálkozásokat automatikusan elfogadja.

A fenti kifejezetten igaz az adatbázis objektumokra, amik feltöltése nagyban fokozott figyelmet követel meg. Ezek elemeinek feltöltése után az elemek bizonyos függvényeit is fel kell építenünk, mindhez felül kell írni a Provider, Expression és ElementType elemeket, valamint a GetEnumerator metódust is. Sok esetben speciálisan a Find metódust is felül kell írni.

Ezek mellett bizonyos tesztekben determinisztikusan megadjuk az adatbázis bizonyos elemeihez tartozó függvények működését is, hogy fókuszban tarthassuk a ténylegesen tesztelt működési elemeket erre találhatunk egy példát a következő ábrán. A képek megalkotása is speciális törődést igényel.

Fent látható az ellenőrző adatok és a kép létrehozása, valamint a Find függvény előre leírása, hogy a kívánt adattal térjen vissza.

4. Összefoglalás

Összességében a Car Service Manager applikáció egy remek eszköz minden olyan felhasználónak, aki szeretné műhelyében leegyszerűsíteni az időpontfoglalás menetét, vagy be szeretne kerülni egy megbízható központi rendszerbe, amiben az egyes számú cél az ügyleteink egyszerű, érthető lebonyolítása.

Az alkalmazás három rétegű Adat-Vezérlés-Nézet (MVC) architektúrája biztosítja a könnyű fejlesztetheit, így az alap funkciók könnyen kibővíthetők további elemekkel, oldalakkal.

A felület fő célja átláthatóság és egyszerűség, színhasználata és tervezési módszerei egységesek, érthetőek, de működésében tartalmaz rengeteg olyan funkciót, ami segítségünkre válhat adatelemzéshez, valamint saját problémáink áttekintéséhez, így a felhasználó soha nem kell elveszve érezze magát. A Razor Page technológia rugalmasságát használva dinamikus módon készítettem jól kinéz, letisztult, funkcionális oldalakat.

A vezérlés egyszerű, törekedünk a műveletek több elemre bontására, a beérkező adatok ellenőrzésére, és a betöltés során átadott adatok minimalizálására. Ezen belül megoldottam a Pdf-ek és Excel táblázatok generálását, valamint az email-es értesítések azonnali, vagy beütemezett kiküldését.

A modell és az adatbázis segítségével könnyen kezeli a beütemezett üzeneteket, a felhasználókat és adataikat, az oldal funkcionalitásához elengedhetetlen adatok eltárolását, módosítását, valamint törlését. Bár alapjaiban nem túl összetett az adatbázis, így is jól össze vannak hangolva az elemek, itt gondolhatunk a foglalások helyhez és személyhez kötöttségére, vagy a helyszínekhez tartozó dolgozók nyilvántartására.

A felhasználókezelés biztonságos, a fiók fölött teljes kontrollt biztosít a felhasználónak. A jogosultságok kiosztása a felületen belülről is megoldható. A jogosultsághoz tartozó funkciók csak akkor jelennek meg, ha az adott jogosultsági szintet valóban elérjük.

Az oldal rengeteg technológiát is összehangolt, előszeretettel alkalmaztam a NuGet csomagok előnyeit az érthetőség és bizonyos funkciók eléréséhez.

Összességében egy funkcionalitásában teljes foglалórendszert hoztam létre, ami teljesíti a bevezetőben kijelölt elvárásokat.

5. Fejlesztési lehetőségek

A program jelen állapotában remekül megfelel az alapötlet céljainak, de ez nem azt jelenti, hogy nem javíthatnánk még rajta, nem adhatnánk hozzá még további ötleteket.

Bár jelen formájában az applikáció autószerelő műhelyekre specializálódik, a kinézet átalakításával bármilyen olyan szolgáltatás működését elősegítheti, amihez időpontokat kell foglaljunk megadott helyeken, opcionálisan adott specialistáknál.

A műhelyi munkák elősegítéséhez csatolhatnánk képeket az autónkról, vagy egy ábrán keresztül kiválaszthatnánk, hogy pontosan melyik részen van a baj, feltéve, hogy rendelkezésünkre áll ez az információ.

Bár a fejlesztéshez elegendő volt a localhost szerveren való futtatás, a működést nagyban megkönnyítené a Microsoft Azure [18], vagy egyéb felhő alapú szerverek felhasználása, hogy az adatok bármikor, bárhonnan elérhető lehessen.

6. Forrásjegyzet

- [1] <https://visualstudio.microsoft.com> – A Visual Studio főoldala. [Hozzáférés dátuma: 07/05/2024]
- [2] <https://www.microsoft.com/en-us/sql-server/sql-server-downloads> - A Microsoft SQL Server letöltési oldala. [Hozzáférés dátuma: 07/05/2024]
- [3] <https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16> – Az SQL Server Management Studio letöltési oldala. [Hozzáférés dátuma: 07/05/2024]
- [4] <https://www.geeksforgeeks.org/mvc-design-pattern/> - Ismertető oldal az MVC paradigmáról. [Hozzáférés dátuma: 07/05/2024]
- [5] <https://www.opera.hu/en/> - Az Opera böngésző weboldala. [Hozzáférés dátuma: 07/05/2024]
- [6] https://www.google.com/chrome/?brand=FHFk&gad_source=1&gclid=CjwKCAjwouexBhAuEiwAtW_Zxx3LKYqx2HxC_zlMhJnRMO7lsAMoSGgEnarQ0sVjl9o9tEV7gYTtbxoCjPEQAvD_BwE&gclidsrc=aw.ds – A Google Chrome böngésző főoldala. [Hozzáférés dátuma: 07/05/2024]
- [7] <https://www.chromium.org/chromium-projects/> - A Chromium projekt főoldala. [Hozzáférés dátuma: 07/05/2024]
- [8] <https://learn.microsoft.com/en-us/ef/core/> - Az EntityFrameworkCore ismertető oldala. [Hozzáférés dátuma: 07/05/2024]
- [9] <https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-8.0> – Az ASP.NET Core ismertető oldala. [Hozzáférés dátuma: 07/05/2024]
- [10] <https://www.nuget.org/packages/NToastNotify> - Az NToastNotify NuGet csomag weboldala. [Hozzáférés dátuma: 07/05/2024]
- [11] <https://itextpdf.com> – A iText weboldala. [Hozzáférés dátuma: 07/05/2024]

- [12] <https://www.epplussoftware.com> – Az EPPlus weboldala. [Hozzáférés dátuma: 07/05/2024]
- [13] <https://mimekit.net/docs/html/Introduction.htm> - A MailKit dokumentációs oldala. [Hozzáférés dátuma: 07/05/2024]
- [14] <https://docs.hangfire.io/en/latest/> - A Hangfire dokumentációs oldala. [Hozzáférés dátuma: 07/05/2024]
- [15] <https://learn.microsoft.com/en-us/dotnet/core/testing/unit-testing-with-mstest> - Az MSTest ismertető oldala. [Hozzáférés dátuma: 07/05/2024]
- [16] <https://github.com/devlooped/moq> - A Moq GitHub oldala. [Hozzáférés dátuma: 07/05/2024]
- [17] <https://getbootstrap.com> – A Bootstrap főoldala. [Hozzáférés dátuma: 07/05/2024]
- [18] <https://azure.microsoft.com/en-us> - A Microsoft Azure főoldala. [Hozzáférés dátuma: 07/05/2024]
- [19] <https://learn.microsoft.com/en-us/ef/core/managing-schemas/scaffolding/?tabs=dotnet-core-cli> – Ismertető a Scaffolding funkcióhoz. [Hozzáférés dátuma: 07/05/2024]
- [20] <https://learn.microsoft.com/en-us/aspnet/core/razor-pages/?view=aspnetcore-8.0&tabs=visual-studio> – Ismertető a Razor Page technológiához. [Hozzáférés dátuma: 07/05/2024]
- [21] <https://developers.google.com/maps> - A Google Maps API főoldala. [Hozzáférés dátuma: 07/05/2024]