**Eötvös Loránd Tudományegyetem**

**Informatikai Kar**

**Informatikatudományi Intézet**

**Programozáselmélet és Szoftvertechnológia Tanszék**

Autós Portál

Szerző: Témavezető:

Vass-Horváth Balázs Dr. Gludovátz Attila

Programtervező informatikus BSc. egyetemi adjunktus

**Szombathely, 2024**

# Tartalomjegyzék

Tartalom

[Tartalomjegyzék 2](#_Toc156940887)

[1. Fejlesztői dokumentáció 1](#_Toc156940888)

[1.1 Felhasznált technológiák, programozási nyelvek 1](#_Toc156940889)

[1.1.1 C# - ASP.NET keretrendszer 1](#_Toc156940890)

[1.1.2 MySql 8.0 1](#_Toc156940891)

[1.1.3 HTML 2](#_Toc156940892)

[1.1.4 JavaScript, jQuery 2](#_Toc156940893)

[1.1.5 CSS, Bootstrap 2](#_Toc156940894)

[1.1.6 GitHub 3](#_Toc156940895)

[1.1.7 Model-View Controller (MVC) 5](#_Toc156940896)

[1.2 Felhasznált szoftverek, fejlesztői környezetek 6](#_Toc156940897)

[1.2.1 Visual Studio 2022 6](#_Toc156940898)

[1.2.2 MySql Server 8.0 6](#_Toc156940899)

[1.2.3 MySql Workbench 6](#_Toc156940900)

[1.2.4 GitHub Desktop 7](#_Toc156940901)

[1.2.5 Microsoft Windows 10 7](#_Toc156940902)

[1.2.6 IIS Web Szerver 7](#_Toc156940903)

[1.2.7 Google Chrome 8](#_Toc156940904)

[1.3 Alkalmazásterv 9](#_Toc156940905)

[1.3.1 Témaválasztás, alkalmazás funkcionális leírása 9](#_Toc156940906)

[1.3.2 Tervezett erőforrásigény 11](#_Toc156940907)

[1.4 Alkalmazás felépítése, működése 12](#_Toc156940908)

[1.4.1 Felépítés 12](#_Toc156940909)

[1.4.1.1 Adatbázis réteg 12](#_Toc156940910)

[1.4.1.2 Backend réteg 13](#_Toc156940911)

[1.4.1.3 Frontend réteg 13](#_Toc156940912)

[1.4.2 Frontend 13](#_Toc156940913)

[1.4.3 Backend 13](#_Toc156940914)

[1.4.4 Adatbázis, tárolt adatok, adatvédelem 14](#_Toc156940915)

[1.4.4.1 Adatbázis terv 15](#_Toc156940916)

[1.5 Beüzemelési javaslat 16](#_Toc156940917)

[1.5.1 MySql szerver telepítése, beüzemelése 16](#_Toc156940918)

[1.5.2 Program összekötése az adatbázissal 17](#_Toc156940919)

[1.5.3 ASP.NET Core Runtime letöltése, telepítése 19](#_Toc156940920)

[1.5.4 IIS Express webszerver beüzemelése 20](#_Toc156940921)

[1.5.5 Webhely létrehozása, project exportálása (Deploy) 20](#_Toc156940922)

[1.5.6 Microsoft Azure felhőszolgáltatás, és lehtőségei 23](#_Toc156940923)

[2 Felhasználói dokumentáció 24](#_Toc156940924)

[2.1 Az oldal meglátogatása 24](#_Toc156940925)

[2.2 Alkalmazás működése, funkciók 24](#_Toc156940926)

[2.2.1 Autentikáció 24](#_Toc156940927)

[2.2.1.1 Regisztráció 25](#_Toc156940928)

[2.2.1.1.1 Végfelhasználó 25](#_Toc156940929)

[2.2.1.1.2 Szolgáltató 26](#_Toc156940930)

[2.2.1.1.3 Gyártó 27](#_Toc156940931)

[2.2.1.1.4 Sikeres regisztrációt követően, email cím megerősítés 27](#_Toc156940932)

[2.2.1.1.5 Lehetséges hibák regisztráció során 28](#_Toc156940933)

[2.2.1.2 Bejelentkezés 29](#_Toc156940934)

[2.2.1.3 Elfelejtett jelszó 30](#_Toc156940935)

[2.2.1.4 Kijelentkezés 32](#_Toc156940936)

[2.2.2 Fióktípusok, jogosultságok, jármű jogviszonyok 33](#_Toc156940937)

[2.2.3 Felső menüsor, navigálás 33](#_Toc156940938)

[2.2.4 Általános, és felhasználói funkciók 34](#_Toc156940939)

[2.2.4.1 Profil 34](#_Toc156940940)

[2.2.4.1.1 Felhasználó fióktípus 35](#_Toc156940941)

[2.2.4.1.2 Szolgáltató fióktípus 36](#_Toc156940942)

[2.2.4.1.3 Jelszó módosítás 36](#_Toc156940943)

[2.2.4.1.4 Státuszok, jogosultságok 37](#_Toc156940944)

[2.2.4.2 Jármű rögzítése a rendszerbe 38](#_Toc156940945)

[2.2.4.3 Saját járművek áttekintése 40](#_Toc156940946)

[2.2.4.4 Járműrészletező, járműkezelés 40](#_Toc156940947)

[2.2.4.4.1 Tankolás rögzítés 41](#_Toc156940948)

[2.2.4.4.2 Egyéb kiadás rögzítése 42](#_Toc156940949)

[2.2.4.4.3 Jogviszonykezelés 43](#_Toc156940950)

[2.2.4.4.4 Járműhöz rögzített információk 44](#_Toc156940951)

[2.2.4.5 Jármű eladásra kínálása 45](#_Toc156940952)

[2.2.4.6 Feladott hirdetés módosítása, törlése 47](#_Toc156940953)

[2.2.5 Adminisztrátor specifikus funkciók 48](#_Toc156940954)

[2.2.6 Szerviz specifikus funkciók 48](#_Toc156940955)

[2.2.7 Kereskedő specifikus funkciók 48](#_Toc156940956)

[2.2.8 Gyártó specifikus funkciók 48](#_Toc156940957)

# Fejlesztői dokumentáció

## Felhasznált technológiák, programozási nyelvek

### C# - ASP.NET keretrendszer

A Microsoft által fejlesztett .NET keretrendszer napjaink egyik legelterjedtebb, legismertebb, leggyakrabban használt keretrendszere. Népszerűsége betudható többek között annak, hogy mára már platformtól függetlenül is fejleszthetünk benne alkalmazásokat, valamint a nyelv fejlesztői hihetetlenül részletes, mintakódokkal teli dokumentációt biztosít a weboldalán. A C# programozási nyelv a .NET keretrendszer része. Fontos kiemelni, hogy teljes mértékben objektum orientált nyelvről beszélünk, melynek alapjai a C++, és Java nyelvek. Mára már elérhető a nyelv 11.0-s verziója is, valamint a keretrendszer 7.0-s verziója. A hozzá tartozó legismertebb, leggyakrabban használt fejlesztői környezetként a Microsoft által fejlesztett Visual Studiot szokták emlegetni. Fontos kiemelni, hogy rengeteg kiegészítő, úgynevezett „Nuget Package” könyvtár érhető el hozzá, mely lényegesen leegyszerűsítheti a munkánkat. A legismertebb csomagok közé tartozik például az adatbáziskezelést leegyszerűsítő „Entity Framework”, vagy a „Newtonsoft JSON” elnevezésű package, mely elősegíti a JSON formátumú adatok kezelését. Webes alkalmazások készítéséhez az ASP.NET-et használhatjuk, amely napjainkra rengeteg féle alkalmazás létrehozását támogatja (Pl.: Web API, Razor Web, MVC Web App), de akár olyan népszerű Frontend keretrendszerekkel is társíthatjuk, mint például a React JS, vagy az Angular.

### MySql 8.0

A MySql az Oracle Corporation által fejlesztett SQL alapú relációs adatbázis szerver. A MySql mára az egyik legelterjedtebb adatbázis kiszolgáló olyan okok miatt, mint például a nyílt forráskód, a megbízhatóság, a stabilitás, valamint az, hogy platform független, tehát elérhető többek között Windows, Linux, MacOS X, FreeBSD, és szinte minden további operációs rendszeren. Az adatbázis kezeléséhez szükségünk van továbbá egy adatbázis kezelő szoftverre, mint például a MySql Workbench. MySql adatbázis kezelő szoftverek is szinte minden operációs rendszerre léteznek, még Androidra, vagy IOS-re is. Alternatívájaként érdemes megemlíteni az MSSQL-t, amely hasonló hatékonysággal képes ellátni az adatbázis feladatokat.

### HTML

A HTML, avagy Hyper Text Markup Language egy weboldalak készítésére szolgáló leíró nyelv. A HTML jelenleg legfrissebb verziója a HTML 5. A HTML önmagában nem alkalmas komplex weboldalak létrehozására azon tulajdonsága miatt, hogy csak a weboldal megjelenéséért felel. A weboldalak formázásáért a CSS felelős, míg a kliens oldalon futó funkciókat jellemzően JavaScript segítségével oldhatjuk meg. Weboldalak esetén a szerveroldali funkciók kiszolgálására napjainkban már rengeteg nyelv biztosít lehetőséget, de a legismertebbek talán a PHP, a C# (ASP.NET keretrendszer), Java (Spring keretrendszer), Python, Node JS.

### JavaScript, jQuery

A JavaScript napjaink egyik legelterjedtebb weboldalakon használt script nyelve. Fontos tulajdonsága, hogy kliens oldalon fut le a kód, melyet a böngésző futtat, azaz a felhasználó által is kiolvasható. Ezen tulajdonságának betudható, hogy komplex alkalmazás fejlesztésére önmagában nem alkalmas, mivel nem futtatható szerver oldalon. Az elnevezése kisség megtévesztő lehet, mivel bár a szintaktikája kissé hasonló a Java-hoz, de semmilyen kapcsolatban nincs a két nyelv. A jQuery egy JavaScript könyvtár, melynek népszerűsége betudható annak, hogy gyorsan, egyszerűen használható, és rengeteg hasznos funkciót, eseménykezelést tartalmaz, melyeket így már JavaScriptben nem kell nekünk elkészíteni. Rengeteg publikus, más programozók által készített jQuery kód érhető el az interneten, melyekkel például bemeneti mezőket, és egyéb, a felhasználó számára látható elemeket tehetünk látványosabbá, használhatóbbá. A jQuery használatához script-ként be kell importálnunk azt az oldalunkra, viszont ez alapból elérhető az ASP.NET projektekben, így ezzel már nem kell foglalkoznunk. Le is tölthetjük, valamint hivatkozhatunk online elérhető verziójára is. Az online verzió mellett szól az, hogy naprakész, viszont hátránya lehet a rendelkezésreállás. További előnye lehet, hogy igény esetén Ajax segítsége mellett egyszerűen használhatjuk API-val.

### CSS, Bootstrap

A CSS, vagy teljes nevén Cascading Style Sheets egy stílusleíró nyelv, mely főleg weboldalak esetén használatos. Célja, hogy a weboldalunk tartalmát és kinézetét elválassza egymástól. A CSS segítségével több módon is lehetőségünk van az oldalunk kinézetének csoportosítására. Hivatkozhatunk egy tag azonosítójára (id), nevére (name), vagy akár típusára is. Továbbá lehetőség van egy adott elem közvetlen, beágyazott stílus módosítására is, ám ez nem ajánlott, ugyanis sokkal olvashatatlanabb kódot eredményez, és nem utolsó sorban nem tudjuk több helyen is használni. A .NET keretrendszer által létrehozott projektekhez automatikusan társul a Bootstrap aktuális verziója. A Bootstrap egy nyílt forráskódú, ingyenes CSS keretrendszer. Alternatíva híján napjainkban a legelterjedtebb front-end keretrendszer. Első körben weboldalak kinézeteinek kialakításához használatos. Napról-napra növekszik a Bootstrap alapú sablonok száma, melyeket web designerek készítenek. Található ezek között ingyenes, valamint fizetős verzió is, de lehetőségünk nyílik saját sablon létrehozására is. Továbbá jelentős, hogy könnyen személyre szabható, és átlátható, erre azonban nem minden esetben van szükség, mivel alapból található benne a legtöbb webfejlesztés során használatos elemre (Például gombok, formok, beviteli mezők) megfelelő kinézet. Ezen kinézeteket leggyakrabban úgy használhatjuk fel, hogy az adott elemre alkalmazzuk a számunkra szükséges CSS osztályt. A Bootstrap használata mellett a legnagyobb érv a reszponzivitás. Reszponzív oldalról akkor beszélhetünk, amikor az azt megjelenítő kijelző méretétől, felbontásától, arányaitól függetlenül az alkalmazás úgy jelenik meg, hogy minden eleme megfelelően látszódjon, valamint olvasható és használható maradjon. Az előzőleg említett tulajdonság az egyik legfontosabb elvárás napjainkban a weboldalaknál, mivel a felhasználók egyre változatosabb eszközöket használnak változatos kijelző specifikációkkal kezdve a nagyfelbontású monitoroktól, az átlagos/nagyfelbontású mobiltelefonokon, okostelevíziókon, táblagépeken keresztül az alacsony felbontású okosórákig.

### GitHub

A GitHub a Microsoft egy leányvállalataként a verziókezelés, verziókövetés lehetőségét biztosítja. A diplomamunkám elkészítése közben számomra fontos volt, hogy szükség esetén az otthoni számítógépemen kívül máshonnan is elérhessem, fejleszthessem a kódomat. Ezen felül lehetőséget biztosít számunkra, hogy egy esetleges hibás fejlesztést követően visszaállítsuk a kódot egy régebbi verziójára, és nem utolsó sorban egy esetleges adatvesztés következtében is vissza tudom nyerni a diplomamunkámat. Segítségével nyomon követhetem, hogy miként fejlődik a program, mikor milyen fejlesztéseket, újításokat építek bele. Csoportos munkák esetén további létfontosságú funkciók is megnyílnak előttünk. Lehetőséget biztosít arra, hogy az általunk létrehozott projektet munkatársaink is elérhessék, megtekinthessék, módosíthassák jogosultságuktól függően. Ebben az esetben gyakran külön branch-et, úgynevezett munkaasztalt kapnak a fejlesztők, és a saját módosításaikat oda töltik fel, míg a fő munkaasztalt csak arra külön kijelölt személyek érik el, ők egyesítik a fejlesztők munkáit. Ez a lehetőség nagyvállalati környezetben elengedhetetlen. Bár legfőképpen forráskódjainkat tároljuk itt, de lehetőség van például dokumentáció, Wikipédia, integrációs könyvtárak (LIB-ek), és még rengeteg virtuális adat tárolására, nyomon követésére. A GitHub továbbá lehetőséget nyújt számunkra kisebb weboldalak tárolására, valamint kódjaink publikálására, és még megannyi hasznos funkcióra, melyeket szinte a végtelenségekig lehetne sorolni. Alternatívaként rengeteg lehetőséget felsorolhatnék, de talán a legismertebb az azonos alapokra épülő GitLab, valamint rengeteg cég mára már saját Git alapú verziókövető rendszert üzemeltet, és használ.

### Model-View Controller (MVC)

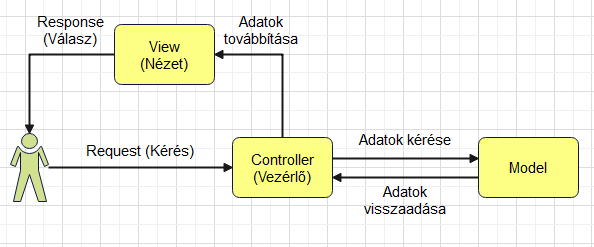
A Model-View-Controller, vagy rövidítve MVC napjaink egyik legnépszerűbb, és leghasználgatóbb (jellemzően webes) program tervezési mintája. Célja, hogy az alkalmazásunkat három fő komponensre bontjuk, melyek a modell, a nézet (view), és a vezérlő (controller). Legfőbb előnyei többek között a párhuzamos fejlesztési lehetőség, vagy éppen a modularitás. Az egyes rétegek feladatai:

* Modell: Jellemzően az alkalmazás által használt adatok strukturált (osztályokba szervezett) tárolásáért, eléréséért felel.

Esetünkben az alkalmazás tartalmaz Adatbázis modelleket (DBModels), melyek az adott modellhez tartozó adatbázis-tábla szerkezetét írja le, és ennek segítségével Entity Frameworkön keresztül könnyen összekapcsolhatjuk az alkalmazásunkat és az adatbázisunkat.

Ezen túl az alkalmazás tartalmaz Request, és Response modelleket, melyek a Nézet- és vezérlő rétegek közötti rendszerezett adatcserét biztosítja.

* Nézet: A nézet réteg felel az alkalmazásunk megjelenéséért, valamint a felhasználóval történő „kapcsolattartásért”. A nézeten jelenítjük meg az adatokat a felhasználó számára, valamint ezen keresztül küld kéréseket a felhasználó a Vezérlő rétegnek (Pl.: egy gomb megnyomásának hatására adatokat küldünk a szervernek).
* Vezérlő: Mint az a nevéből is adódik, az alkalmazásunk vezérléséért felel. Ez a (webes alkalmazások esetén szerveroldali) réteg felel azért, hogy feldolgozza a nézet rétegtől kapott (felhasználó által kiváltott) kéréseket, valamint a kérésekre valamilyen válasszal reagáljon. Ez a válasz webes alkalmazás esetén lehet egy elérési út (Nézet) visszaadása, vagy adatok visszaadása a megjelenítő számára, esetleg valamilyen hibakód.



1. ábra MVC felépítés

## Felhasznált szoftverek, fejlesztői környezetek

### Visual Studio 2022

A Visual Studio a Microsoft több programozási nyelvet támogató, valamint az évek során folyamatosan fejlődő fejlesztői környezete. Lehetőséget biztosít többek között C#, C++, F#, Visual Basic, Python és még néhány további nyelven történő alkalmazás fejlesztésre. Ezeken túl több Leíró (Pl.: XML, HTML), és Script (Pl.: JS) nyelvet, kliens oldali keretrendszert (Angular, React), valamint JSON formátumot is képes felismerni. A fejlesztői környezet első verzióját 1997-ben mutatták be, viszont az évek során rengeteg verzió jelent meg (jellemzően 3-4 évente), napjainkban a 2022-es a legfrissebb. A .NET keretrendszerben történő fejlesztés először a 2002-es verzióban jelent meg.

### MySql Server 8.0

A MySql Server 8.0 verziója az egyik lehető legjobb választás, ha valamilyen komplex alkalmazáshoz szeretnénk adatbázis kiszolgálót. Legnagyobb előnye, hogy platformfüggetlenség mellett ingyenesen elérhető szinte minden elterjedt operációs rendszerre. Az adataink tárolása mellett lehetőséget biztosít kényszerek használatára is, mely segítségével tovább csökkenthetjük annak esélyét, hogy hibás adatok kerüljenek be az adatbázisunkba. További lehetőségünk van adatbázis programozásra is, mint például triggerek, és tárolt eljárások készítése.

### MySql Workbench

A MySql Workbench az Oracle által fejlesztett grafikus MySql adatbáziskezelő szoftver. Előnye, hogy ingyenes, és ezen túl Windows, Linux, valamint OS X rendszerekre is elérhető. Továbbá célszerű választás lehet azért is, mivel az Oracle cég fejleszti, akik a MySql fejlesztői is, ezáltal feltehető, hogy folyamatosan naprakész verziót használhatunk. Az alkalmazás telepítését a MySql Server 8.0 telepítője is felajánlja, de külön is letölthetjük a mysql hivatalos oldaláról. Segítségével nem csak az adatbázisainkat kezelhetjük, de a MySql szerverünket is, továbbá rendelkezik például teljesítmény elemzéssel, és még sok más beépített funkcióval. Hátrányaként említeném, hogy a felhasználói felülete számomra elmarad a legtöbb fizetős adatbáziskezelő rendszerével szemben.

### GitHub Desktop

A GitHub Desktop a GitHub saját fejlesztésű szoftvere. Célja, hogy a GitHub-on kezelt munkáinkat könnyebben, átláthatóbban elérhessük. Segítségével megkönnyíthetjük az életünket, ugyanis rengeteg GitHub CLI-ben kiadott parancsot válthatunk ki csupán néhány kattintással. Az alkalmazás elérhető Windows, valamint MacOS rendszerekre is. Lehetőséget biztosít a verziókövetés nyújtotta előnyök szinte minden lehetőségének kihasználására, mint például a projectünk klónozása, módosítások feltöltése, vagy épp korábbi verziók visszaállítása. Bár mára már a Visual Studio is biztosítja a verziókövetést valamilyen szinten, funkcionalitása elmarad a GitHub Desktop apptól. Itt kiemelném, hogy a GitHub rendelkezik mobilos alkalmazással is, mely elérhető Android, és IOS eszközökre is.

### Microsoft Windows 10

A fejlesztéshez használt eszközeimen Microsoft Windows 10 operációs rendszert használok, mely napjaink egyik legelterjedtebb operációs rendszere. Folyamatos biztonsági, és funkcionális frissítéseknek köszönhetően garantálja a felhasználói élményt, és a naprakész biztonságot. Mára elérhető a Windows 11 operációs rendszer is, ám mivel egy viszonylag friss rendszerről beszélünk, ezért még előfordulhatnak kompatibilitási, vagy stabilitási problémák.

### IIS Web Szerver

Az alkalmazás futtatásához IIS Web Szervert használtam, mely a minden újabb Windows operációs rendszeren alapból elérhető, csupán aktiválni kell. Segítségével minden ASP.NET alatt készített alkalmazást futtatni tudunk, és mint minden más fejlett web szerver, ez is lehetőséget ad számunkra minden fontos biztonsági funkció használatára. Napjainkban a legnépszerűbb web szerver kiszolgálók közé tartozik, ezért a folyamatos frissítések garantálják a stabilitást és a naprakész biztonságot. Tovább könnyíti a dolgunkat, hogy az aktiválását követően a windows frissítésekkel együtt a web szerverünk is automatikusan kapja a frissítéseket. A fejlesztés folyamán is találkoztam már a szerver frissítésével, és semmilyen problémát nem okozott a futtatott alkalmazások üzemelésében, ezért ettől sem kell tartanunk.

### Google Chrome

A Google Chrome egy ingyenesen elérhető webböngésző, mely többek között elérhető Windows, Linux, MacOS, Android, és IOS rendszerekre is. A felhasználók előszeretettel használják, mert felhasználóbarát kezelőfelülettel rendelkezik, és végtelenül egyszerűen használható. További pozitívum, hogy számos bővítmény érhető el hozzá, hátrányként viszont érdemes megemlíteni a konkurens böngészőkhöz képest valamivel magasabb memóriahasználatot.

## Alkalmazásterv

### Témaválasztás, alkalmazás funkcionális leírása

Nagyon sok időt töltöttem azzal, hogy a diplomamunkámnak szánt alkalmazás témáját kitaláljam. Rengeteg ötletem támadt, ám végül egy olyan témájú app mellett döntöttem, amely az informatika után talán a legközelebb áll a szívemhez: az autózás/autóipar. Már kisgyerek koromban is rengeteget foglalkoztam a témával, ám akkor még csak játékként tekintettem rá: Autós kártyák, Hot Wheels, később lego autók, autóversenyzős számítógépes játékok, és persze a kedvenc mesefilmem is a verdák című animációs film volt. Nem is volt kérdés, amint lehetett, 17éves koromban megszereztem a jogosítványt, azóta pedig csak még tovább nőtt az autózás iránti rajongásom, szinte nem telik el úgy nap, hogy ne ülnék kormány mögé. Mivel a választott témakörben szintén rengeteg lehetőség rejlik, főleg napjainkban, amikor az ipar, és az autózás is hatalmas változásokon megy keresztül, ismét elkezdett pörögni az agyam: Milyen célt szolgáljon az alkalmazásom? – Önvezető autó szimuláció? Autóversenyzős játék?

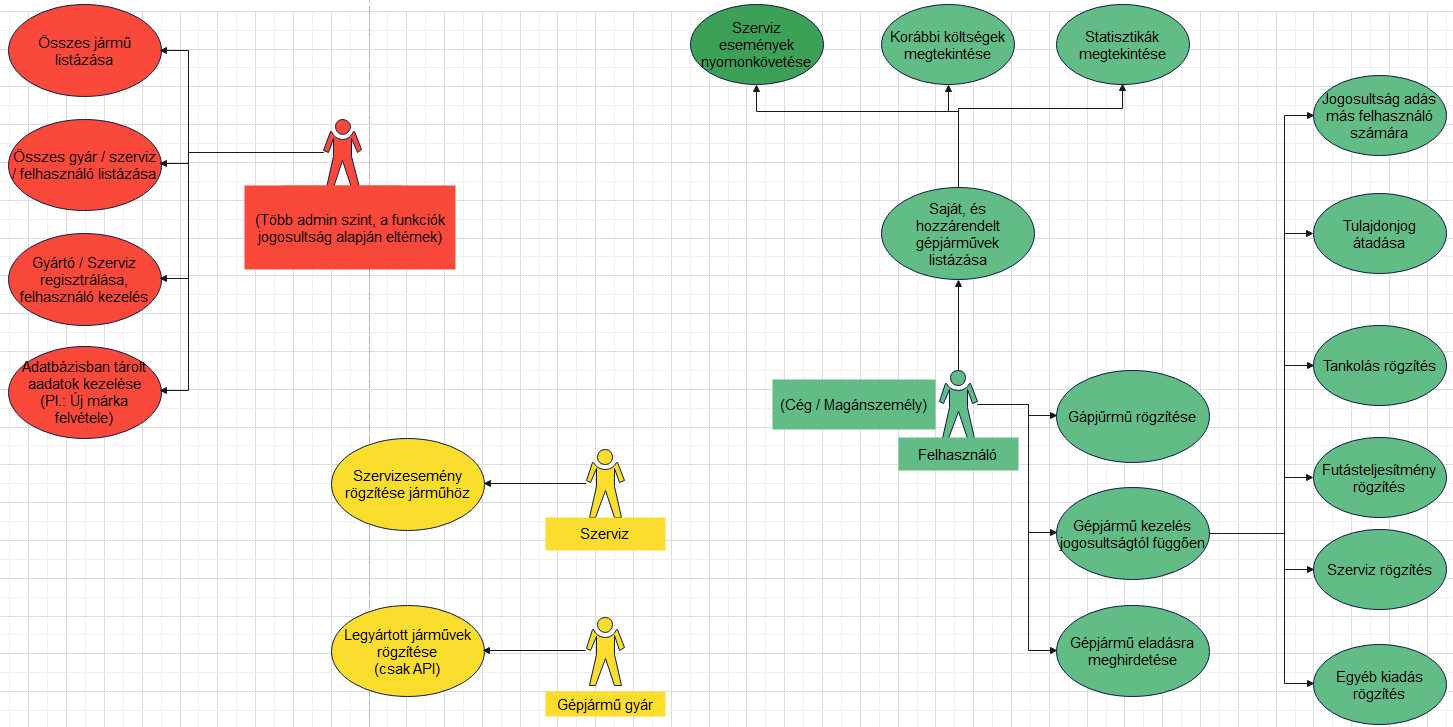
Végül egy ezeknél sokkal célszerűbb ötletem támadt: Egy gépjármű nyilvántartó alkalmazás. Mint azt már írtam, az autózás, és ezáltal az autóipar hatalmas nehézségeken, és változásokon megy át napjainkban. Éppen, hogy kezdett lecsengeni a COVID-19 nevű világjárvány, egy újabb nehézséget kell áthidalni: Kitört az Orosz-Ukrán háború, mely hatalmas inflációt eredményezett világszinten. Tapasztalható, hogy jelentősen megváltoztak az emberek autózási- és autóvásárlási szokásai is ebben a válságközeli helyzetben: A legtöbben sokkal költséghatékonyabban próbálnak közlekedni, és a használtautó piac egyre jobban maga alá szorítja az újautó piacot.

Az általam kigondolt alkalmazás célja, hogy elősegítse a költséghatékony gépjármű üzemeltetést. A felhasználók számára lehetőséget biztosít, hogy rögzítse a gépjárművének költségeit, mint például a tankolás, a szervizek, vagy bármilyen további, a felhasználó által megadott költség. Ezeken túl lehetőséget biztosít, hogy nyomon kövessük a gépjármű használati szokásainkat úgy, hogy rögzíteni tudjuk az aktuális futásteljesítményét, illetve a tankolásokat.

Mindazonáltal gépjárműveket nem csak magánszemélyek használnak, sőt a járműhasználat csupán töredékét teszik ki magánszemélyek. Az alkalmazás éppen ezért sokkal inkább a vállalkozásokat célozná meg: Lízing cégek, Személy- és áruszállítással foglalkozó cégek, futárszolgálatok, vagy bármilyen cégautó(ka)t tartó vállalkozások. Lehetőségünk van korlátlan számú gépjárművet rögzíteni a rendszerbe, melyeknek teljes életciklusát lekövetné az alkalmazás a vásárlástól egészen az eladásig. Miután a céges autókat jellemzően nem csak egy személy vezeti, ezért az alkalmazás lehetőséget biztosít arra, hogy egy járművet a tulajdonoson kívül több felhasználóhoz is kirendeljen, akik jogosultságuktól függően (Tulajdonos, Üzembentartó, Sofőr) érhetik el az alkalmazás bizonyos funkcióit a járműre nézve.

Az alkalmazás főbb tervezett funkciói:

* Autógyártó cégek számára: API-n keresztül lehetőséget biztosít a legyártott gépjárművek rögzítésére a rendszerbe.
* Autószervizek számára: A náluk szervizelt gépjárművekre rögzíteni tudják az elvégzett szervizek adatait (szerviz időpont, elvégzett munka, költségek, egyéb megjegyzések/észrevételek)
* Cégek/Vállalkozások számára: A cég gépjárműveinek kezelése, nyilvántartása; Jármű hozzárendelése felhasználó(k)hoz; Új jármű(vek) rögzítése; Jármű(vek) eladásra meghirdetése, használt járművek keresése, eladásra kínált jármű rögzített adatainak megtekintése; Sikeres eladás esetén az új tulajdonosnak átadhatja a járművet, aki így tovább tudja vezetni a jármű eseményeit.
* Magánszemélyek számára: A cégekhez hasonló funkciók korlátozott elérése (Pl.: Limitált számban rögzíthet csak járműveket, vagy épp limitált számban rendelheti felhasználóhoz azt), a hozzá rendelt járművek kezelése jogosultságtól függően (Tulajdonos; Üzembentartó/Másodlagos tulajdonos; Sofőr).
* Admin jogosultság: Természetesen az alkalmazás rendelkezik jogosultságkezeléssel, ugyanis bizonyos esetekben előfordulhat, hogy adminisztrátori közbenjárásra van szükség. Ilyen például a kereskedők / szervizek regisztrációjának megerősítése, vagy a gyártók regisztrálása. Ezen túl még rengeteg felhasználó-támogató funkcióval rendelkezik, mint például a járművek tulajdonjogának átírása, valamint felhasználó támogató szerepet is betöltenek.



2. ábra Követelmény specifikáció

### Tervezett erőforrásigény

Az alkalmazás üzemeltetéséhez szükséges erőforrás-igény megbecsülése talán az egyik legnehezebb feladat. Fejlesztés során ügyeltem az optimális erőforrás-felhasználásra, ezzel is csökkentve a szükséges hardware erőforrást. Mivel webes alkalmazásról beszélünk, ezért az első, és legfontosabb kitétel a stabil, nagy sávszélességű dedikált internetkapcsolat. Nem elhanyagolható továbbá a folyamatos rendelkezésreállás se, ezért javasolt szünetmentes tápegység használata is. Az alkalmazás zár körű (akár céges környezetben) történő használatához elegendő csupán egy átlagosnak nevezhető szervergép (Windows Server operációs rendszerrel számolva):

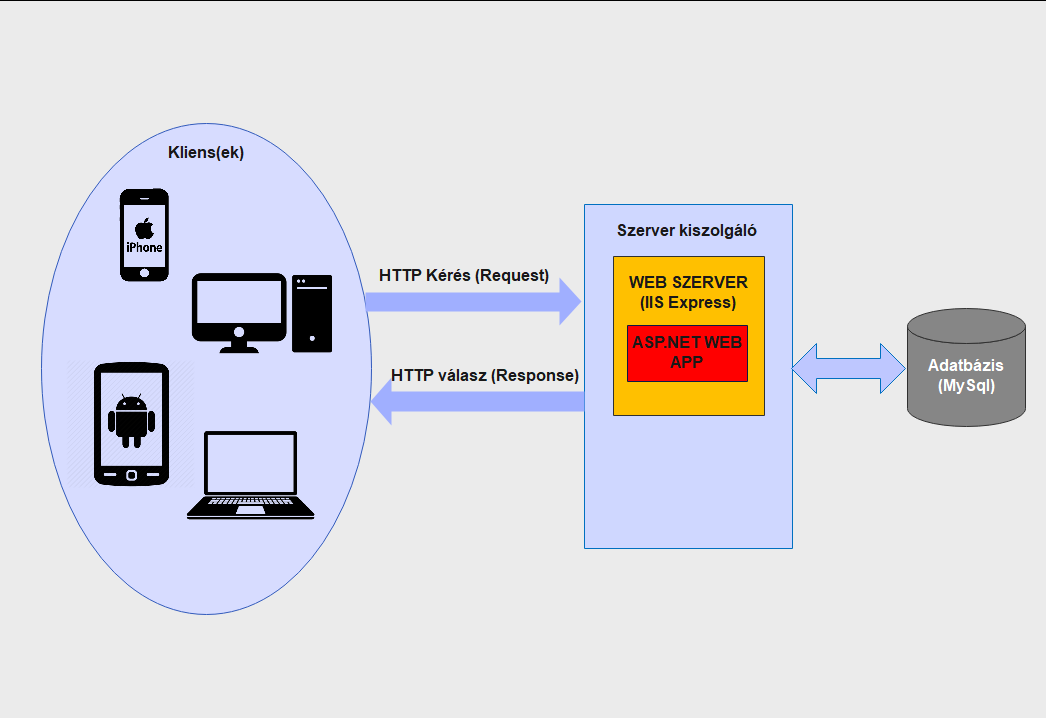
* RAM: 8GB belső memória
* CPU: min. 1.8GHz 4mag
* Tárhely: 25GB belső tárhely

Mindazonáltal amennyiben az alkalmazást rendeltetésének megfelelően, globális környezetben szeretnénk használni, abban az esetben egy komplett szerver farmra lehet szükségünk.

## Alkalmazás felépítése, működése

### Felépítés

Az alkalmazást három fő rétegre bonthatjuk. Ezek az Adatbázis, a Backend (szerveroldal), és a Frontend (kliensoldal). A felhasználók ezek körül csupán a frontendet, azaz a kliensoldali réteget látják, és érik el közvetlenül.



3. ábra Alkalmazás felépítése

A rétegek feladatai:

#### Adatbázis réteg

Az adatbázis feladata, hogy az alkalmazás által használt adatok strukturált, biztonságos tárolását biztosítsa. A biztonság garantálása érdekében a felhasználó közvetlenül semmilyen módon nem férhetnek hozzá ehhez a réteghez. A kapcsolatot egy köztes réteg, a backend biztosítja. Természetesen az adatbázis a szerveroldali rétegek közé tartozik.

#### Backend réteg

A Backend (azaz szerveroldali réteg) fő feladata, hogy a kliens, és az adatbázis közötti biztonságos összeköttetést garantálja. Segítségével ellenőrizni tudjuk, hogy az adatbázisunkba csak megfelelő adatok kerülhessenek, valamint hogy a felhasználók csak a jogosultságaiknak megfelelő tartalmakat érhessék el.

#### Frontend réteg

A frontend, azaz (kliensoldali) felhasználói réteg feladata, hogy a felhasználók számára is értelmezhető, átlátható, és kezelhető módon megjelenítse a szükséges adatokat, valamint lehetőséget biztosítson, hogy a felhasználó kényelmesen kéréseket küldjön a szerver felé (Pl.: adat beszúrás), valamint a szervertől kapott választ megjelenítse a felhasználónak.

### Frontend

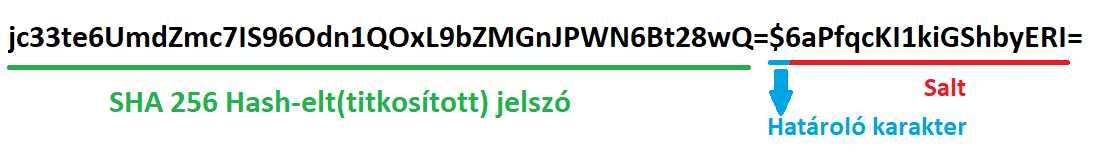
Az alkalmazás megjelenítéséért a kliens oldal felel. A tartalomért HTML, a megjelenésért CSS, azon belül is Bootrstrap keretrendszer, míg a kliens oldali funkciók működéséért JavaScript, valamint a legismertebb JS könyvtár, a JQuery felel. Az ASP.NET keretrendszer lehetőséget biztosít számunkra az MVC tervezési minta használatára, melyet jellemzően Razor Web-es technológiával szokás társítani. Az alkalmazás felépítésének köszönhetően lehetőségünk van egy projecten belül fejleszteni a Kliens- és Szerver oldalt is úgy, hogy mégis elkülönítjük a rétegeket, azaz biztosítjuk a párhuzamos fejlesztési lehetőséget, és az átláthatóságot. Szimplán néhány kattintással létrehozhatunk egy új „Nézetet” (oldalt) az alkalmazásunkhoz. Fontos kiemelni, hogy a nézeteinkbe csupán az adott oldal „body” részének tartalmát kell megírnunk, ugyanis az alkalmazás minden lapja egy úgynevezet Layout-ra (sablonba) kerül bele, mely tartalmazza az alkalmazás vázát (Pl.: Navbar, Boorstrap, és JQuery könyvtárak importálása). Amennyiben egy nézethez kliensoldali funkciókat (scripteket) szeretnénk hozzáadni, abban az esetben a kódot érdemes a nézeten belül „Scripts” section-be rakni, melyet az alkalmazásunk a sablonban, a body végén fog „behúzni”. Hasonló a helyzet a formázás esetében is, itt „CSS” section-be írjuk a kódjainkat, melyet az alkalmazásunk a head rész megfelelő helyére fogja behúzni.

### Backend

Az alkalmazás szerveroldali rétegét egy C# nyelven. ASP.NET keretrendszer alatt készült alkalmazás szolgálja ki. A szerveroldal legfőbb komponensei a controllerek, avagy vezérlők. Minden adatbázis művelet keresztül megy a vezérlőnkön, ugyanis a kliensek a controller felé intéznek kéréseket, melyre a szerverünk választ ad. Annak érdekében, hogy ne sértsük meg a szoftverfejlesztési elveket (Pl.: SOLID) érdemes a lehető legjobban elkülöníteni a komponenseket, épp ezért az alkalmazásban minden fő tevékenységi körre külön controller van (Pl.: Auth, Admin, User, Api). Az adatbázissal történő kommunikáció Entity Frameworkon keresztül történik, melynek segítségével nem kell folyamatosan adatbázis lekéréseket írnunk, hanem elég egy táblastruktúrát leíró modellt készítenünk, majd DBSet-eken keresztül már el is érhetjük az adatbázisunkat.

### Adatbázis, tárolt adatok, adatvédelem

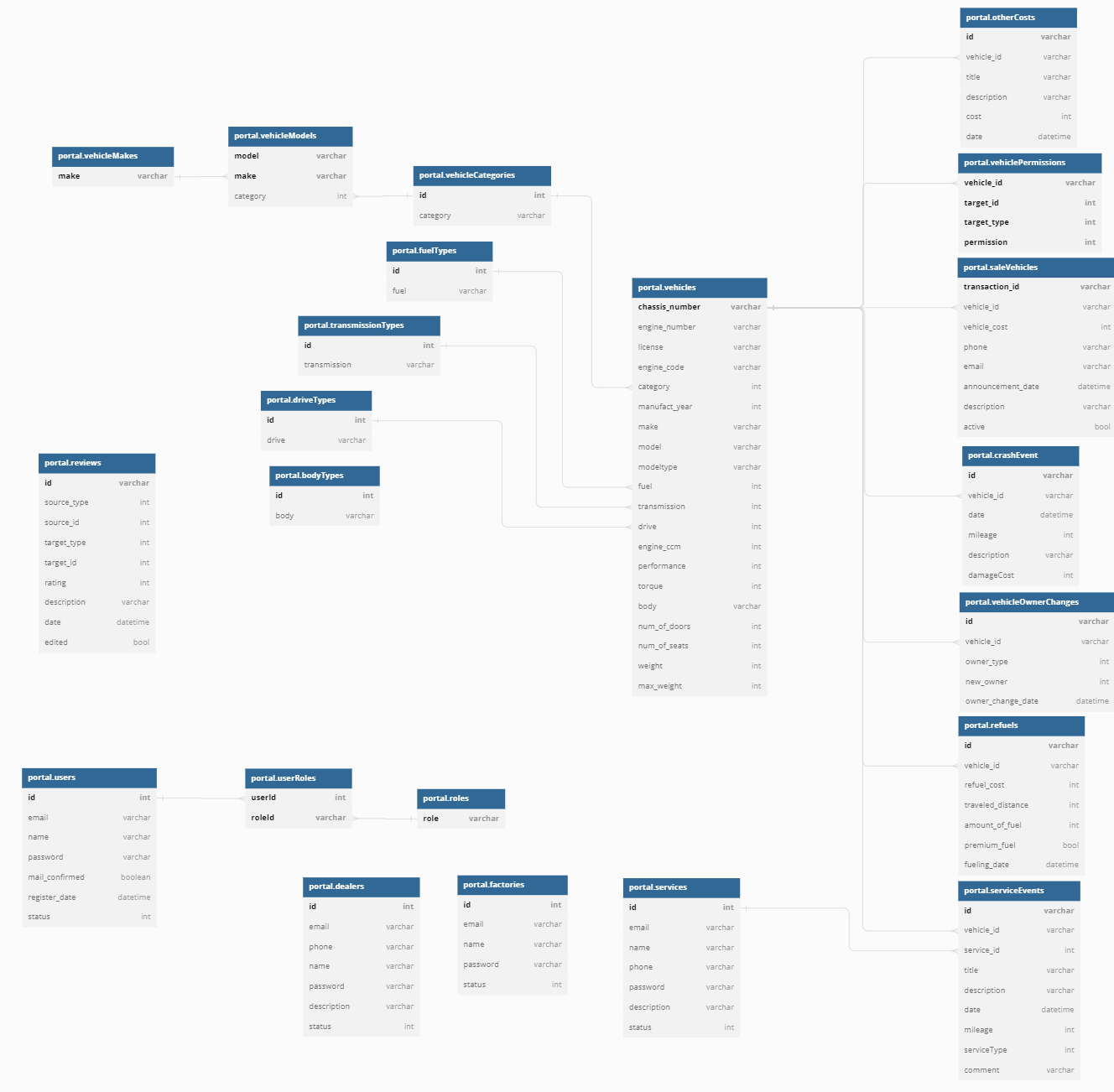
Az adatbázis struktúra kialakításakor fontos szempont volt az optimális felépítés, valamint a megfelelő tábla-kapcsolatok kialakítása is annak érdekében, hogy elkerüljük az adatintegritási problémákat, bár az alkalmazás kliens, és szerver oldali adatellenőrzést is végez az adatmanipulációs műveletek előtt. Mivel az alkalmazás rendeltetésszerű használata során hatalmas mennyiségű adat kerülhet eltárolásra, ezáltal fontos volt, hogy azok eltárolása optimális, átgondolt legyen. Egy hibásan megtervezett, vagy elkészített adatbázis esetén nem csak a végtelenségekig lassíthatjuk az alkalmazásunkat, de akár egy idő után az egész appot használhatatlanná teheti. Az alkalmazás minden felhasználó típusának a jelszavát saltozva, SHA-256 titkosítással tárolja. Az SHA-256 titkosítás során a bevitt jelszóból egy 256 bit hosszúságú karakter sorozat keletkezik. A hashelési algoritmusnak köszönhetően így egy biztonságos jelszó feltörése bruteforce megoldással napjainkban még egy szuperszámítógép számára se valósítható meg belátható időn belül, természetesen visszafejtésre (napjainkban) szintén nincs lehetőség. A saltozás (sózás) célja, hogy esetleges két azonos karakter sorozat esetén is eltérő legyen a hash-elt érték, melyet úgy érünk el, hogy a hashelés előtt egy véletlenszerűen generált (esetünkben 10-15 közötti véletlen hosszúságú) karaktersorozatot (ez lesz az ún. salt) fűzünk a titkosítani kívánt karaktersorozathoz. Természetesen a saltot szintén el kell tárolnunk, hogy később (esetünkben bejelentkezéskor) a bevitt jelszóhoz újra hozzáfűzhessük, és ezáltal a generált hash megegyezzen az tárolttal. A saltot esetünkben a jelszó mögött tároljuk el ’$’ határolókarakterrel elválasztva. Az alábbi képen egy minta látható az eltárolt jelszóról:



4. ábra Jelszó tárolás

#### Adatbázis terv

Az alkalmazás adatait tároló MySql adatbázis pillanatnyi felépítése az alábbi ábrán látható:



5. ábra Adatbázis terv

## Beüzemelési javaslat

Az általam készített alkalmazás egy webes alapú rendszer, ezáltal többféleképpen üzemeltethetjük (Pl.: helyi számítógépen, szerver gépen, felhőben), melyeknél a beüzemelés módja eltérő lehet. Én a windows operációsrendszer alatti beüzemelésre fogok kitérni. A beüzemeléshez szükség lehet valamennyi rendszergazdai ismeretre, tapasztalatra.

### MySql szerver telepítése, beüzemelése

Töltsük le a MySql Server lehetőleg legújabb (jelenleg 8.0) támogatott verzióját az oracle hivatalos oldaláról, majd haladjunk végig a telepítőn. A telepítés során szükséges lesz, hogy megadjuk majd az úgynevezett ’root’ (admin) felhasználónk jelszavát, mindenképpen jegyezzük meg az itt megadott jelszót, ugyanis ha elfelejtjük, akkor később ezt meg kell változtatnunk MYSQL SHELL-ben, mely szintén némi MYSQL-kezelési ismeretet kíván.

Jelentkezzünk be az adatbázis szerverünkre, az általunk használt adatbázis-kezelő alkalmazással (Pl.: MySql Workbench), a felhasználó nevünk ’root’, míg a jelszó az általunk megadott érték lesz. Hozzunk létre egy új MySql felhasználót az alkalmazásunk számára.

Ezt a „CREATE USER `felhasznalo`@`host` IDENTIFIED BY `password`;” paranccsal tehetjük meg. Természetesen a ’felhasznalo’ helyére az általunk kívánt felhasználó nevet írjuk (célszerű olyant választani, mely utal a felhasználására, Pl.: portal), a ’host’ helyére az alkalmazást futtató kiszolgáló IP címét (Pl.: localhost), mellyel lekorlátozzuk, hogy csak az adott címről lehessen belépni a felhasználóba, míg a ’password’ helyére az újonnan létrehozott felhasználónk jelszava fog kerülni. Jelszó választáskor célszerű megfelelő hosszúságú, kis-és nagybetűket, számokat, valamint speciális karaktereket egyaránt tartalmazó (akár véletlenszerűen generált) jelszót választani.

Az adatbázisunk létrehozását a mellékelt adatbázis forrásfájl tartalmazza, ezáltal elegendő, ha a fájl tartalmát egy az egyben lefuttatjuk. Ekkor létrejön az adatbázisunk ’portal’ néven.

Most adjunk jogosultságot a létrehozott felhasználónknak az adatbázisunkhoz való hozzáférésre. A jogosultság megadásakor célszerű, hogy a felhasználónk az adatbázisunkhoz korlátlanul hozzáférjen (ellenkező esetben az alkalmazás nem fog megfelelően működni). A jogosultság megadásához futtassuk le a „GRANT ALL PRIVILEGES ON portal.\* TO 'username'@'host';” természetesen a ’username’, és ’host’ értékeket itt is cseréljük ki a létrehozott felhasználónk adataira. A parancs lefuttatásának a hatására a felhasználónk teljes hozzáférést (adatok beszúrása, törlése, módosítása; tábla létrehozása, törlése, módosítása stb.) kap a portal adatbázisunk minden táblájához.

Ezzel az adatbázisunk beüzemelésével végeztünk is.

### Program összekötése az adatbázissal

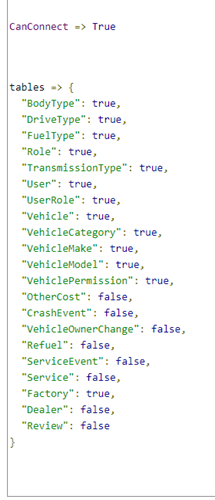
Következő lépésben kössük össze az alkalmazásunkat az adatbázissal. Mivel az alkalmazás rendelkezik saját „config” fájllal, ezért különösen egyszerű dolgunk van. Keressük meg az alkalmazásunk mappáját, majd nyissuk meg az „appsettings.json” fájlt. A fájl szerkesztésére akár még hagyományos jegyzettömböt is használgatunk, mindazonáltal célszerű valamilyen olyan szerkesztőt használni, amely átláthatóan tagolja a szöveges állományt, és még akár systax highligting is rendelkezésre áll. Ilyen alkalmazások lehetnek például a notepad++, vagy a visual studio code, de akár a használt fejlesztői környezetben, a visual studio 2022-ben is módosíthatjuk.

A konfigurációs állományban keressük ki a „ConnectionString”, azon belül pedig az „SQL” részt, majd állítsuk be az adatbázisunk elérhetőségét. Mivel a konfigurációs fájl egy JSON állomány, ezért fontos figyelni annak szintaktikájára is.



6. ábra Alkalmazás-Adatbázis összekötés

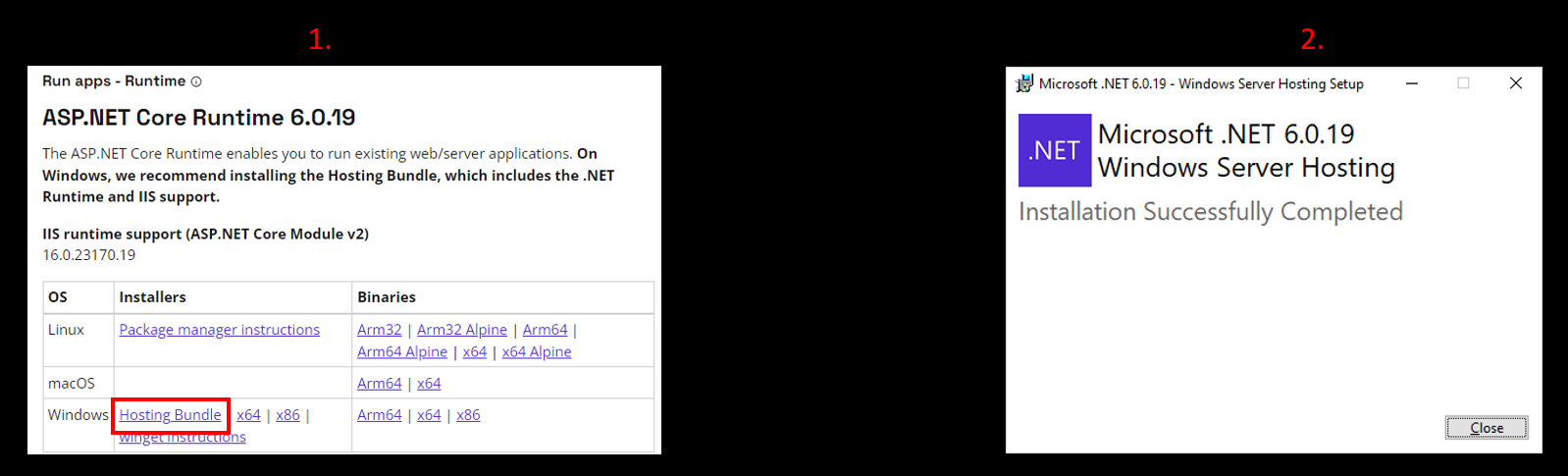
A kapcsolat kialakításának sikerességét valamint a tábláink meglétét úgy ellenőrizhetjük, hogy az alkalmazás elindítását követően az „IP/Dev/CheckDbConnection” elérési útra navigálunk. Ekkor JSON formátumban látni fogjuk az adatbázisunk státuszát (Csatlakozási próbálkozás) a CanConnect alatt, majd az összes táblánkat felsorolva állapot jelzéssel. Természetesen a true jelenti a sikert, ellenben ha false értékel látunk, akkor valamilyen hiba van, ebben az esetben ellenőrizzük az adatbázisunkat, és a kapcsolatot.



7. ábra Adatbázis kapcsolat ellenőrzése

### ASP.NET Core Runtime letöltése, telepítése

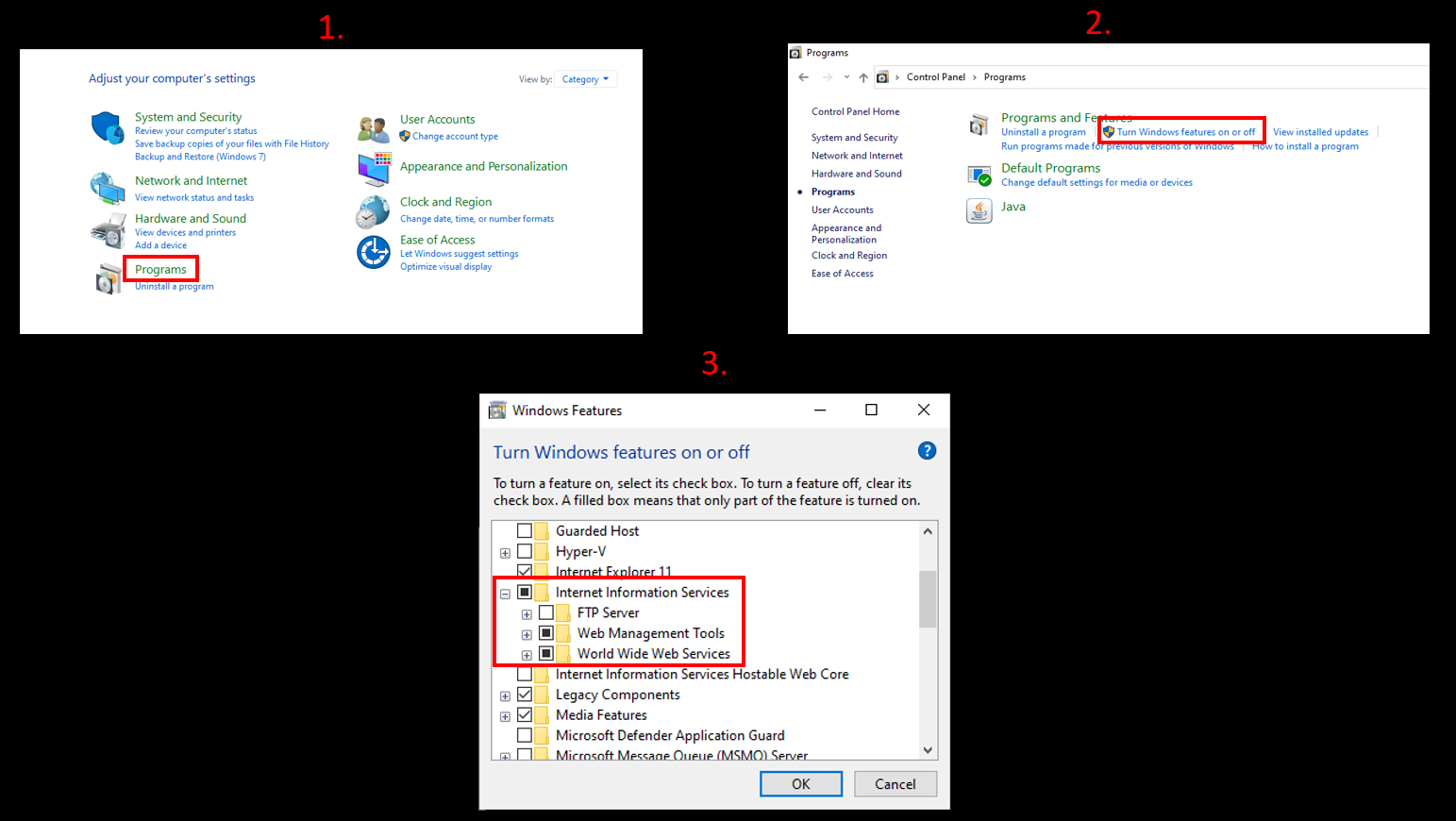
Az alkalmazás futtatásához szükségünk van az ASP.NET Core Runtime (lehetőség szerint) legújabb verziójára, melyet a dotnet hivatalos oldaláról tölthetünk le (<https://dotnet.microsoft.com/en-us/download/dotnet/6.0>). Javasolt a Hosing Bundle verzió letöltése windows alatt, ugyanis ez minden szükséges dolgot feltelepít. Amennyiben Linux, vagy macOS rendszeren szeretnénk futtatni az alkalmazást, akkor sem kell aggódnunk, ugyanis azokra is elérhetőek ugyan itt a szükséges telepítők. Sikeres telepítés után érdemes újraindítani a gépünket annak érdekében, hogy minden módosítást biztosan elvégezzen, és életbe léptessen a rendszerünk.



8. ábra .NET keretrendszer letöltése, telepítése

### IIS Express webszerver beüzemelése

Az IIS Express webszerver alapból elérhető minden Windows10 operációs rendszert futtató számítógépen, mindazonáltal használatához engedélyeznünk kell ezt a szolgáltatást. Ennek legegyszerűbb módja, hogy belépünk a vezérlőpultba, kiválasztjuk a Programok menüpontot, azon belül pedig a Programok és szolgáltatások alatti Windows-szolgáltatások be- és kikapcsolása opció lehetőséget válasszuk. Fontos megjegyezni, hogy rendszergazdai jogosultságra van szükségünk ahhoz, hogy a szolgáltatást engedélyezzük. Ekkor előjön egy ablak. Itt keressük meg az Internet Information Services, illetve az Internet Information Services üzemeltető webmagja lehetőségeket, és a mellettük található négyzet bepipálásával engedélyezzük a szolgáltatásokat. Amennyiben még korábban nem használtuk ezeket, abban az esetben a rendszerünk automatikusan frissítéseket keres hozzá, és letölt minden hozzá szükséges eszközt, amely kis időbe telhet.

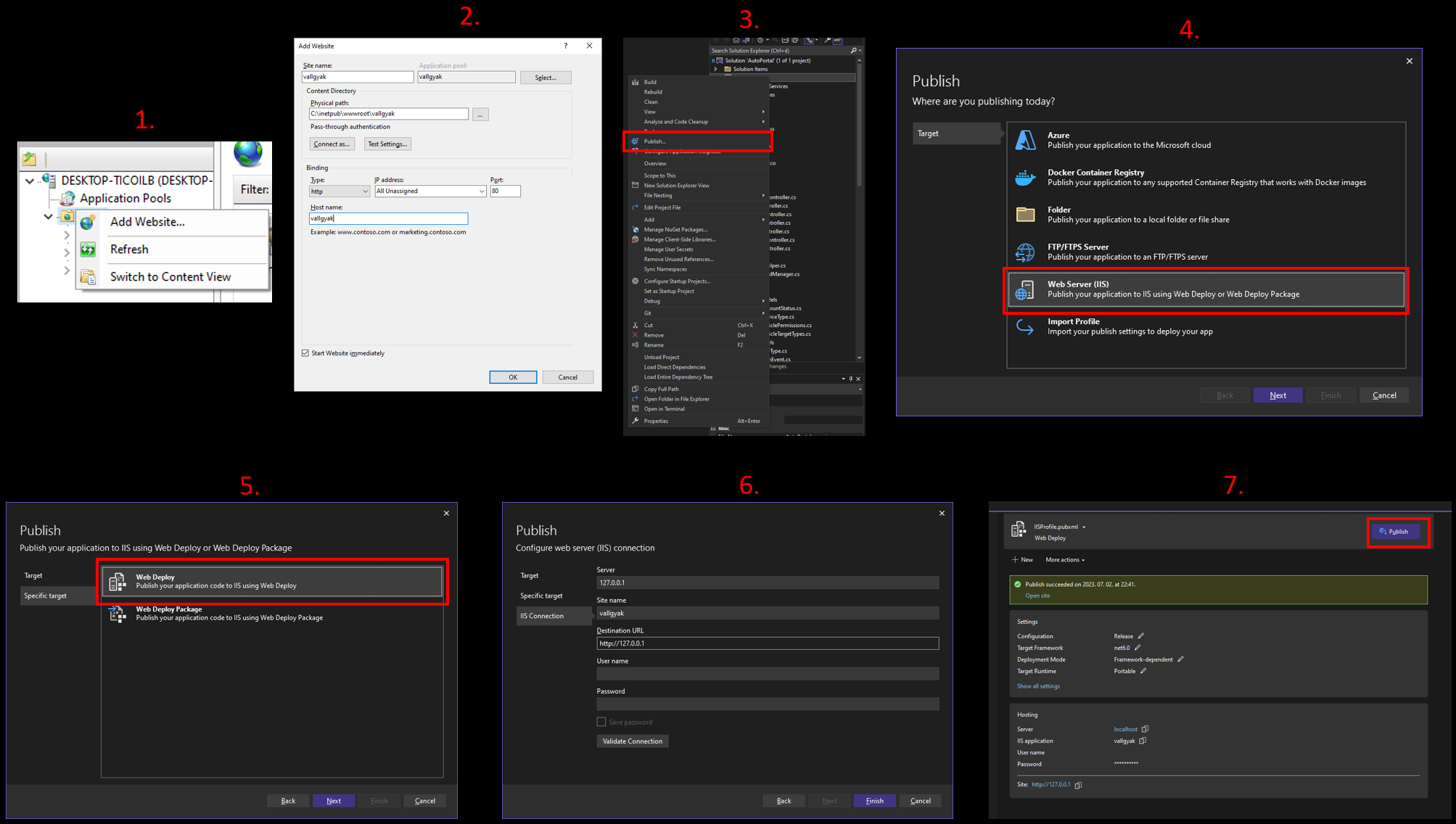


9. ábra IIS Express Webszerver beüzemelése

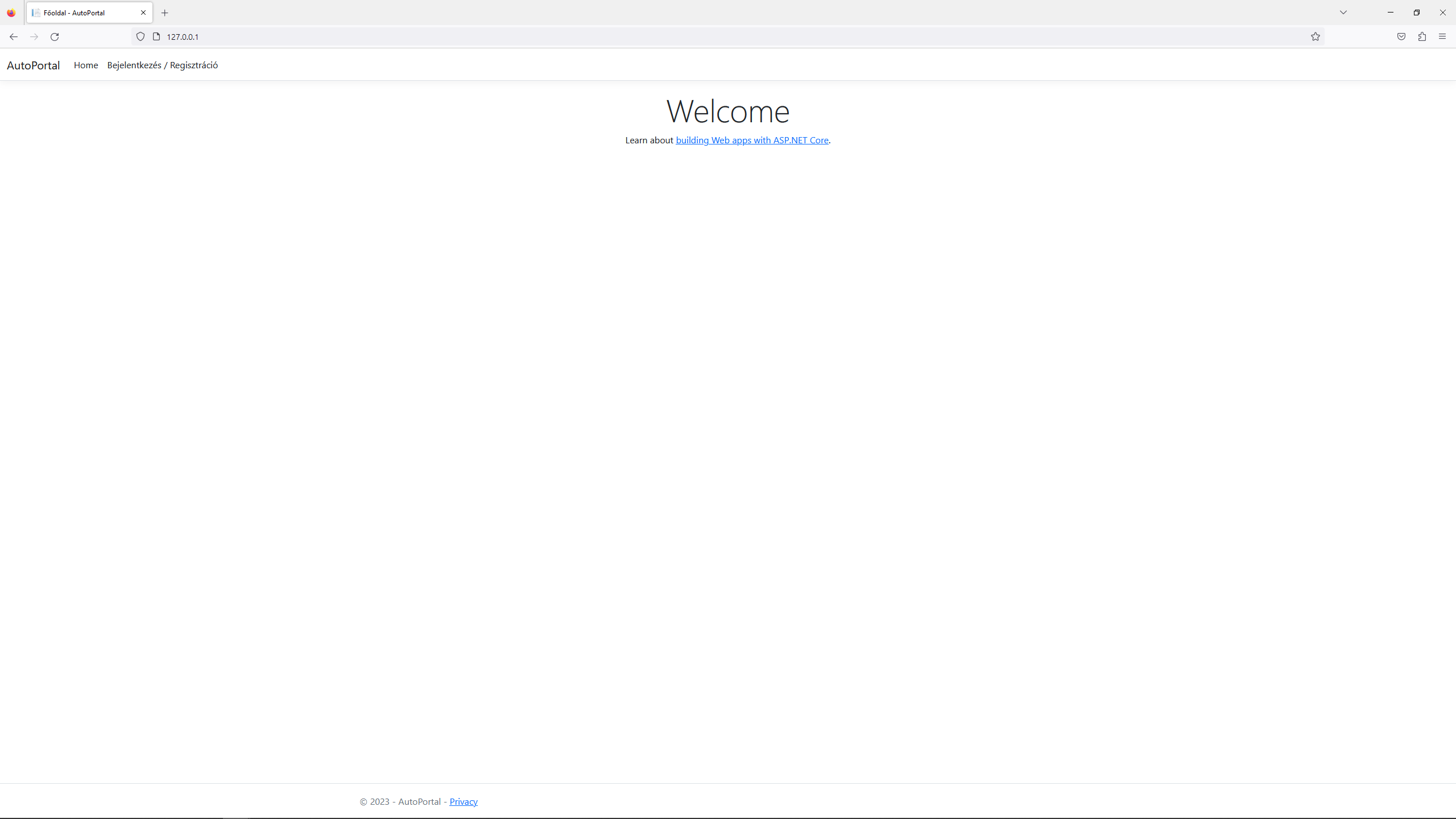
### Webhely létrehozása, project exportálása (Deploy)

Miután aktiváltuk a webszerverünket, indítsuk újra a számítógépünket. Ezt követően elérhetővé válik számunkra egy új alkalmazás, mely „Internet Information Services (IIS) kezelője” névre hallgat. Nyissuk is meg ezt az alkalmazást, amely elérhetővé teszi a szerverünk kezelőfelületét. Baloldalon megjelenik a számítógépünk neve, ezt nyissuk le, hogy megjelenjen a helyek címszóra hallgató opció. Erre kattintsunk jobb egérgombbal, és válasszuk a webhely hozzáadása lehetőséget. A hely neve alatt tetszőleges nevet adhatunk meg, viszont javasolt olyan címet választani, hogy tudjuk, miről van szó. A fizikai elérési úthoz azt a helyet kell kiválasztanunk, ahová az oldalunkat „Telepítjük”. Továbbá itt állíthatjuk be, hogy http, vagy https protokollt szeretnénk használni, valamint azt, hogy mely porton szeretnénk futtatni az oldalunkat. Éles környezetben javasolt a https protokoll használata, ugyanis az jóval biztonságosabb, továbbá mondhatni kötelező az SSL tanúsítvány használata, melyet alapból kapunk a webszerver mellé, így ezt csak ki kell választanunk. Állomásnévhez az oldalunk domain nevét állítsuk be, majd miután mindennel végeztünk kattintsunk az OK gombra. Miután ezekkel végeztünk kattintsunk a baloldali listában a számítógépünket jelző elemre, és a jobboldali menüben kattintsunk az indításra. Miután elindult a webszerverünk a helyek között válasszuk ki a korábban létrehozott webhelyünket, és szintén a jobboldali menüben indítsuk el azt is.

Következő lépésként indítsuk el rendszergazdaként a Visual Studio 2022-t, és nyissuk meg a projektünket. A jobb oldali solution explorer-ben kattintsunk jobb egérgombbal a projectünkre, majd válasszuk a publish opciót. Ekkor megjelenik számunkra egy új ablak, ahol megjelennek a lehetséges exportálási opciók. Esetünkben válasszuk ki a Web Server (IIS) opciót, majd kattintsunk a Next (tovább) gombra. A következő lépésben válasszuk ki a sima Web Deploy opciót, és szintén kattintsunk a tovább gombra. A most megjelenő ablakban kell megadnunk a szerverünk adatait. A Server mezőbe adjuk meg a szerverünk címét (esetünkben localhost), az oldalunk nevét (ennek meg kell egyeznie a webszerveren létrehozott oldal névvel) a Site Name mezőbe, ez esetünkben a vallgyak. A Destination URL, azaz cél elérésiút mezőbe pedig adjuk meg az oldalunk majdani elérési útját (protokoll://IPCÍM). Amennyiben nem helyi, hanem külső szerverre történik az exportálás szükséges felhasználónév, és jelszó megadása is. Miután minden adatot helyesen megadtunk, a Validate Connection gomb megnyomásával ellenőrizhetjük a csatlakozást, majd kattintsunk a Finish gombra. A Publish gombra kattintva sikeres fordítás után fel is kerül az oldalunk a szerverre, melyet az alapértelmezett böngészőben meg is nyit nekünk a Visual Studio.



10. ábra Alkalmazás beüzemelése



11. ábra Beüzemelt alkalmazás

### Microsoft Azure felhőszolgáltatás, és lehtőségei

Az alkalmazás üzemeltetésénél mindenképpen kiemelném a Microsoft által nyújtott Azure felhőszolgáltatást, mely rengeteg lehetőséget biztosít számunkra. Segítségével egy platform alatt elérünk minden olyan szolgáltatást, mely az alkalmazásunk üzemeltetéséhez szükséges. Ilyenek például a web szerver, az adatbázis szerver, vagy éppen a domain név vásárlás. Azonban egy web alkalmazás élet környezetben történő üzemeltetése nem merül ki ebben a néhány lépésben, ugyanis nem szabad megfeledkeznünk például a rendelkezésreállásról, és terhelésmegosztásról sem, ugyanis például application gateway szolgáltatást is igénybe tudunk venni. Ezen szolgáltatás segítségével az alkalmazásunk nem csupán egy, hanem több szerverről szolgálja ki a felhasználókat, így csökkentve a magas terhelés esetén a válaszidőt, valamint csökkentjük annak esélyét, hogy az alkalmazásunk teljes mértékben elérhetetlenné váljon (egy szerver kiesése esetén a többi szerverünk a meghibásodottól függetlenül ki tudja szolgálni a kéréseket, és át tudják vállalni annak szerepét). Természetesen az adatbázisunk kiesése esetén az alkalmazásunk szintén használhatatlanná válik, ezért érdemes arról is folyamatosan mentést készíteni, vagy még biztonságosabb megoldásként két- vagy több párhuzamosan működő adatbázist üzemeltetni, melyeken a tárolt adatok megegyeznek (mindegyiket folyamatosan frissítjük), viszont egyik kiesése esetén az alkalmazásunk azonnal, kiesés nélkül át tud állni egy másikra. Mindazonáltal a Microsoft által nyújtott felhőszolgáltatásoknál a garantált magasszintű rendelkezésreállásnak köszönhetően (akár közel 100%-os) nem kell túlgondolnunk a meghibásodási, és rendelkezésreállási incidensek lehetőségét.

Egy népszerű webes alkalmazást szinte folyamatos támadások érnek (Például DOS, DDOS, POD), melyeket a Microsoft tűzfalai szintén mondhatni 100%-os hatékonysággal tudnak kivédeni megfelelő beállításokkal, ezért ezektől szintén nem kell tartaunk. A folyamatos frissítéseknek köszönhetően mindig naprakész szerver hátteret kapunk.

Mint minden szolgáltatásnál, a minőséget sajnos itt is meg kell fizetni, épp ezért az előnyök után most hátrányként kiemelném az Azure által nyújtott szolgáltatások talán legnagyobb hátrányát, a magas árazást. Bár folyamatosan (heti-havi) rendszerességgel fejlődik, és változik a szolgáltatás, jelenleg nagyon kevés szolgáltatás érhető el ingyen, azok is csak korlátozottan.

## API dokumentáció járműgyárak számára

Járműgyárak számára az alkalmazásunk csak, és kizárólag API-n keresztüli kommunikációt biztosít.

### API elérése

Az alkalmazáshoz tartozó API a <http://localhost:4288/api/> URL alatt érhető el.

### API authentikáció

Az API eléréséhez JWT token authentikáció szükséges. A bejelentkezéshez lásd: [Login](#_Login).

Ezt követően a kérések fejlécéhez kell hozzáadni a tokent az alábbi formátumban:

### API funkciók

|  |  |  |
| --- | --- | --- |
| URL | Method | Rövid leírás |
| [auth/login](#_Login) | POST | Bejelentkezés |
| [factory/addBrandNewVehicle](#_Legyártott_jármű_felvétele) | POST | Új jármű rögzítése |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

#### Login

Az API funkciók eléréséhez authentikáció szükséges. A bejelentkezés POST metóduson keresztül történik az auth/login elérési úton. Paraméterként két értéket várunk:

1. email: string (szöveg) típus, a fiókhoz tartozó email cím
2. password: string (szöveg) típus, a fiókhoz tartozó jelszó

A válasz JSON formátumban érkezik, mely két paramétert tartalmaz:

1. Success: logikai érték, igaz, ha sikerült bejelentkezni, különben hamis
2. Message: string (szöveges) típus, Sikeres bejelentkezés esetén a későbbi azonosításhoz használatos token, ellenkező esetben a hibaüzenet.

Lehetséges visszatérési kódok:

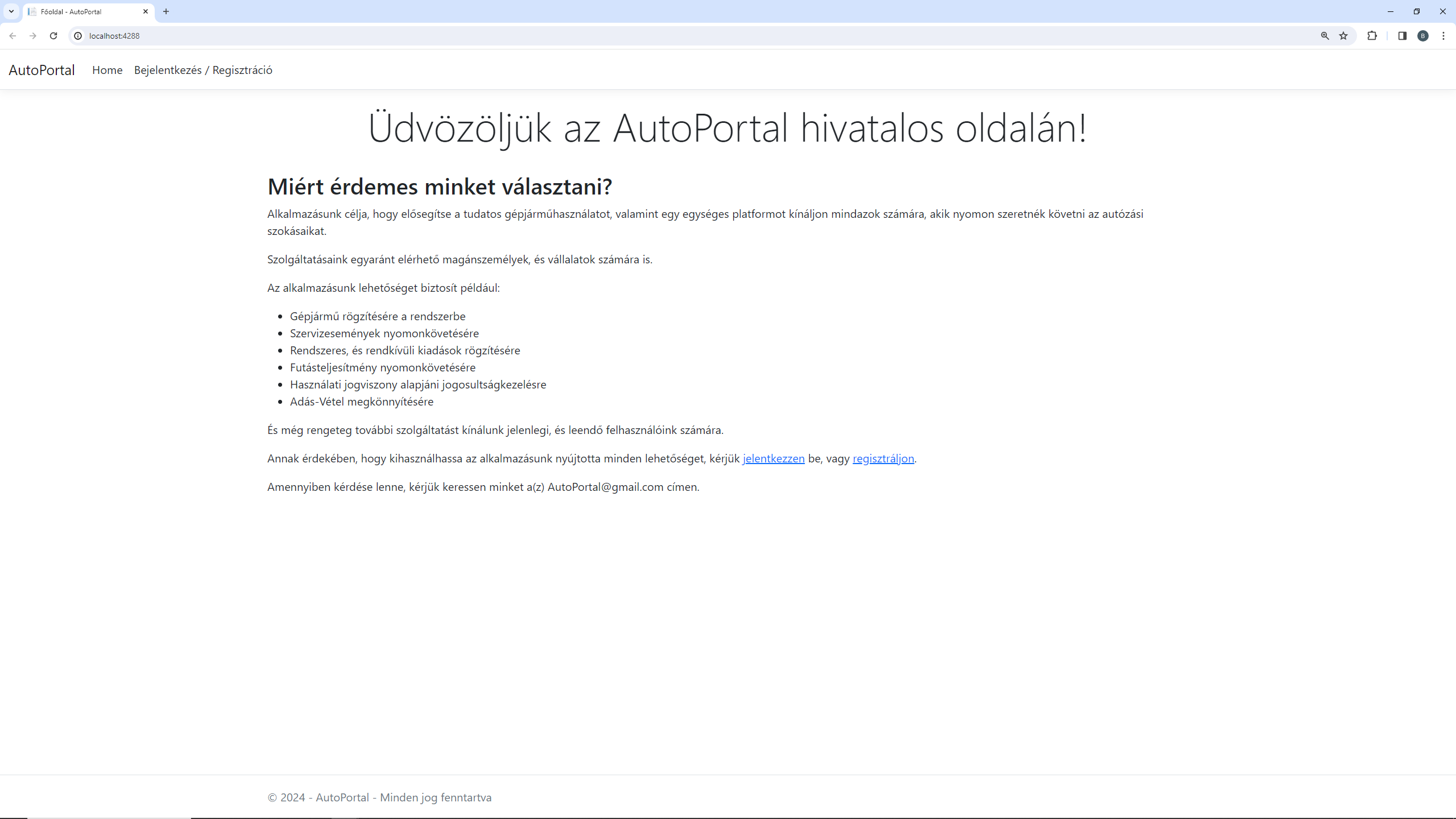
1. 200 (OK) – Sikeres bejelentkezés
2. 400 (Bad Request) – Hibás email, vagy jelszó, vagy hiányos adatok
3. 403 (Forbidden) – A felhasználót adminisztrátor kitiltotta

#### Legyártott jármű felvétele a rendszerbe

# Felhasználói dokumentáció

## Az oldal meglátogatása

Az alkalmazás webes alapokra épül, ezáltal annak használatához szükségünk van egy webböngészőre, valamint internetkapcsolatra. A webes alkalmazások egyik legnagyobb előnyének, a platformfüggetlenségnek köszönhetően a digitális készülékünk típusától (okostelefon, táblagép, laptop asztali pc, okos TV, okosóra…), valamint az azon futó operációs rendszertől (Android, Windows, Linux, MacOS, Tizen…) függetlenül használhatjuk az alkalmazás nyújtotta összes funkciót. A böngészőnk címsorába írjuk be az oldal elérhetőségét, ami lehet IP cím, vagy domain név is.



## Alkalmazás működése, funkciók

### Authentikáció

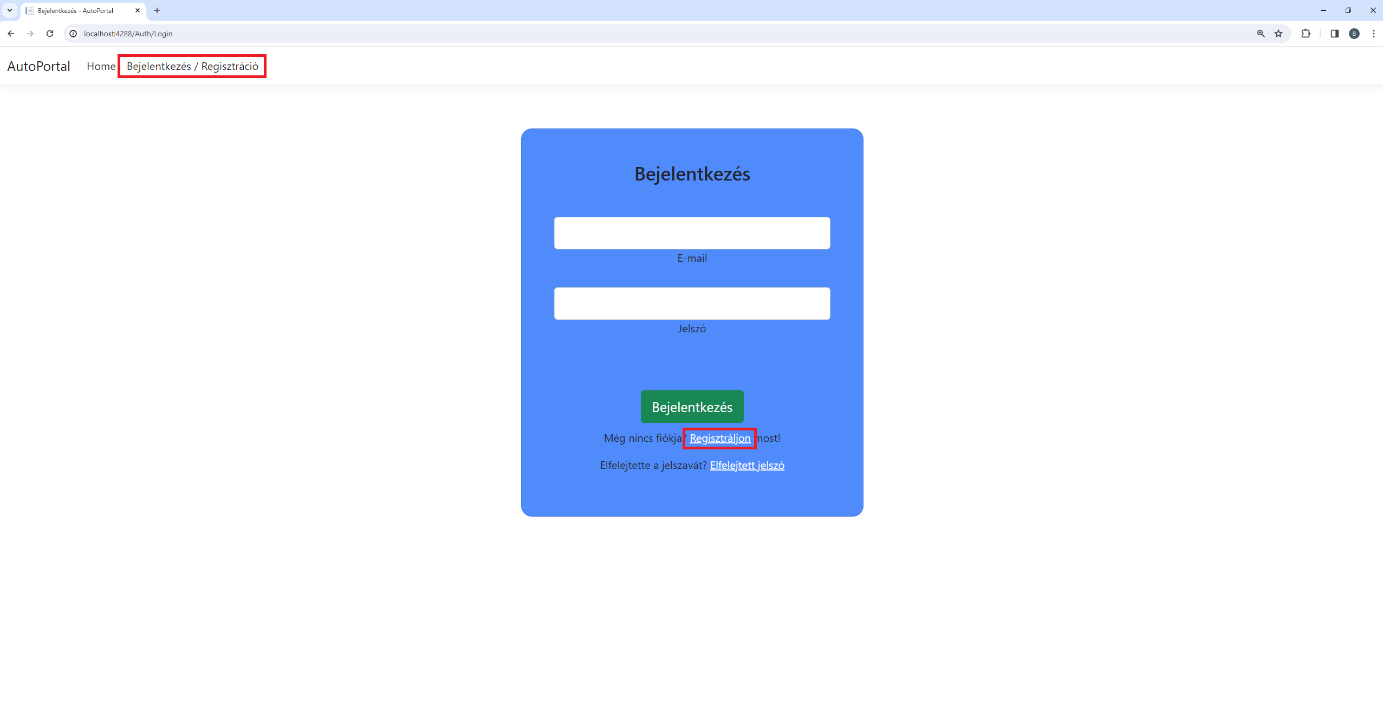
Az alkalmazás rengeteg lehetőséget kínál, ezek többsége azonban bejelentkezéshez kötött. A bejelentkezés legfőbb célja, hogy a felhasználók, és adataik biztonságát garantáljuk, és személyre szabott szolgáltatásokat nyújthassunk, ezért kérjük a felhasználóinkat, hogy az alkalmazás teljesértékű használatához jelentkezzenek be, vagy regisztráljanak. Az alkalmazás jelenleg az alábbi jogosultsági körökben biztosít azonosítás.:

* Gyártók (csak API-n keresztül)
* Kereskedők
* Szervizek
* Végfelhasználók

A végfelhasználói azonosításhoz továbbá adminisztrátori jogosultság is társulhat.

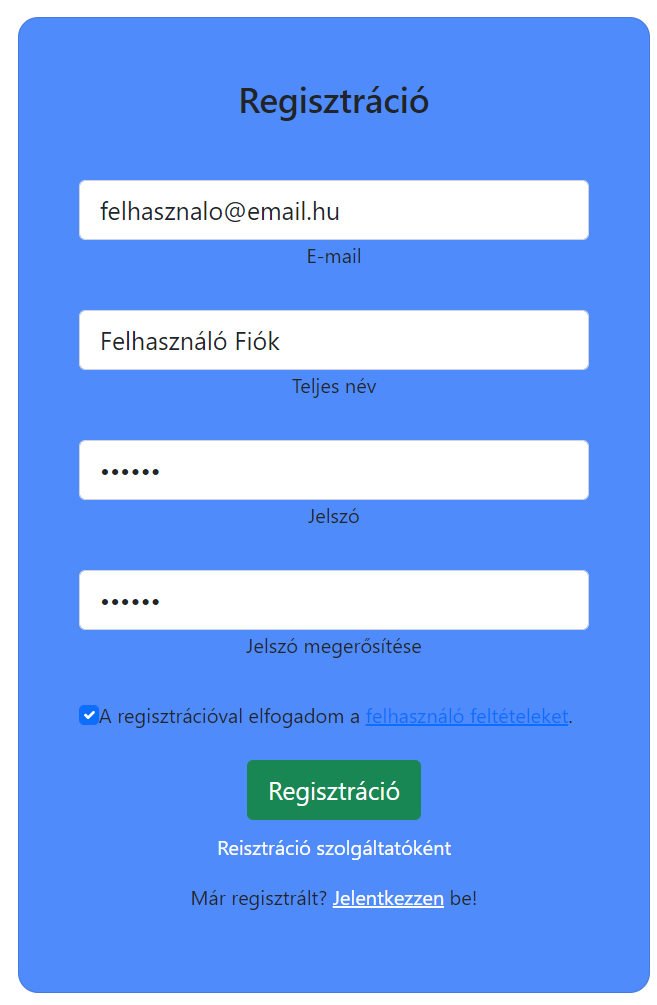
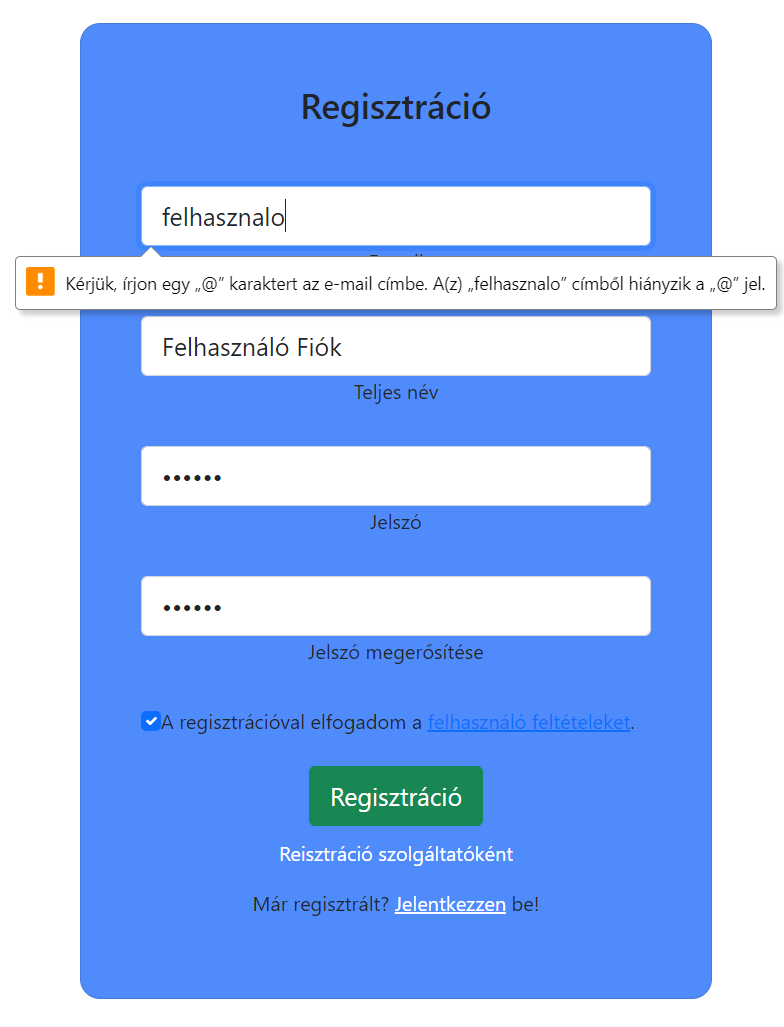
#### Regisztráció

A bejelentkezéshez elengedhetetlen, hogy a felhasználó regisztrálva legyen a rendszerben. A regisztráció menete minimálisan eltér a regisztrált fiók típusától függően. Kérjük, hogy felhasználóink minden esetben valós email címet használjanak, továbbá győződjenek meg róla, hogy korábban még nem regisztráltak az oldalra ugyan azzal az email címmel. A regisztrációs oldal csak, és kizárólag be nem jelentkezett felhasználók számára érhető el a felső menüsáv bal oldalán található „Bejelentkezés / Regisztráció” gombra, majd azt követően a „Regisztráljon” feliratra kattintva.



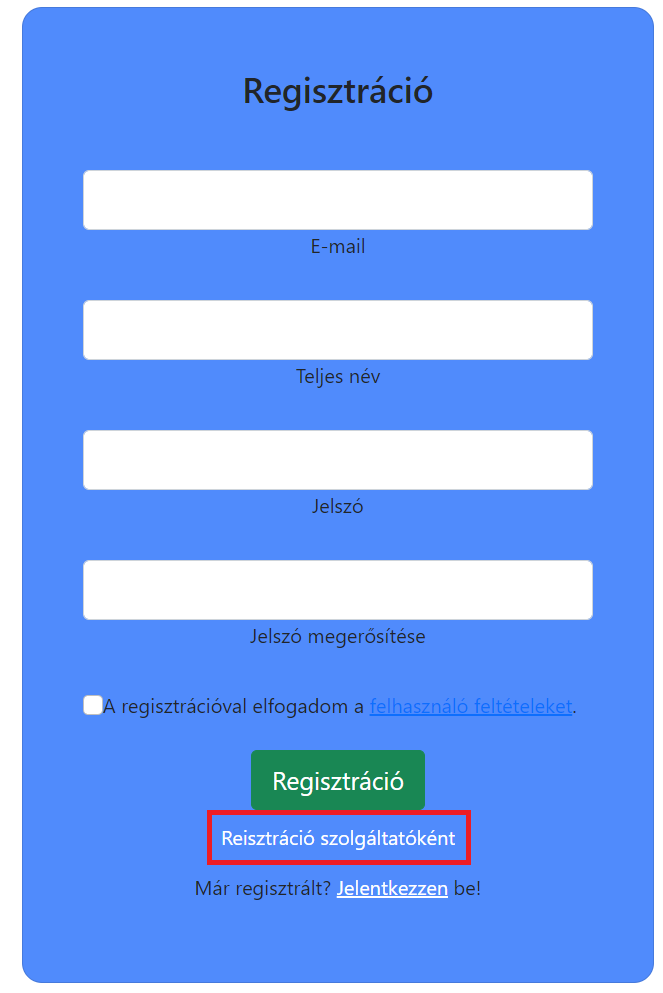
##### Végfelhasználó

Töltsük ki megfelelően a regisztrációs oldalon található mezőket. Ügyeljünk rá, hogy csak valós adatokat adjunk meg, továbbá győződjünk meg róla, hogy a megadott email cím megfelelő, és még nincs regisztrálva az oldalon. A jelszó megválasztásakor igyekezzünk biztonságos jelszót választani, amit máshol nem használunk. Javasolt, hogy tartalmazzon kis, és nagybetűket, számokat és speciális karaktert, valamint kikötés, hogy minimum 6 karakter hosszúságú legyen. Az alkalmazás a lehetőségekhez mérten ellenőrzi a bevitt adatok helyeségét, például az email cím formátumát, valamint a jelszó hosszát. A regisztrációhoz kötelező elfogadni a felhasználói feltételeket. Miután minden adatunkat leellenőriztük kattintsunk a regisztráció gombra.



##### Szolgáltató

Szolgáltatóként történő regisztrációhoz kattintsunk a „Regisztráció szolgáltatóként” lehetőségre, majd töltsük ki megfelelően az itt megjelenő mezőket. Szolgáltatói regisztrációnál különösen fontos, hogy adatainkat megfelelően adjuk meg, ugyanis szolgáltatói fiókot csak, és kizárólag adminisztrátori jóváhagyás után lehet használni, melyet csak és kizárólag megfelelően létrehozott, jól dokumentált szolgáltatók kapnak meg. Szolgáltatóként regisztrációt követően további elérhetőségi adatokat állíthatunk be a cégünkkel kapcsolatosan.

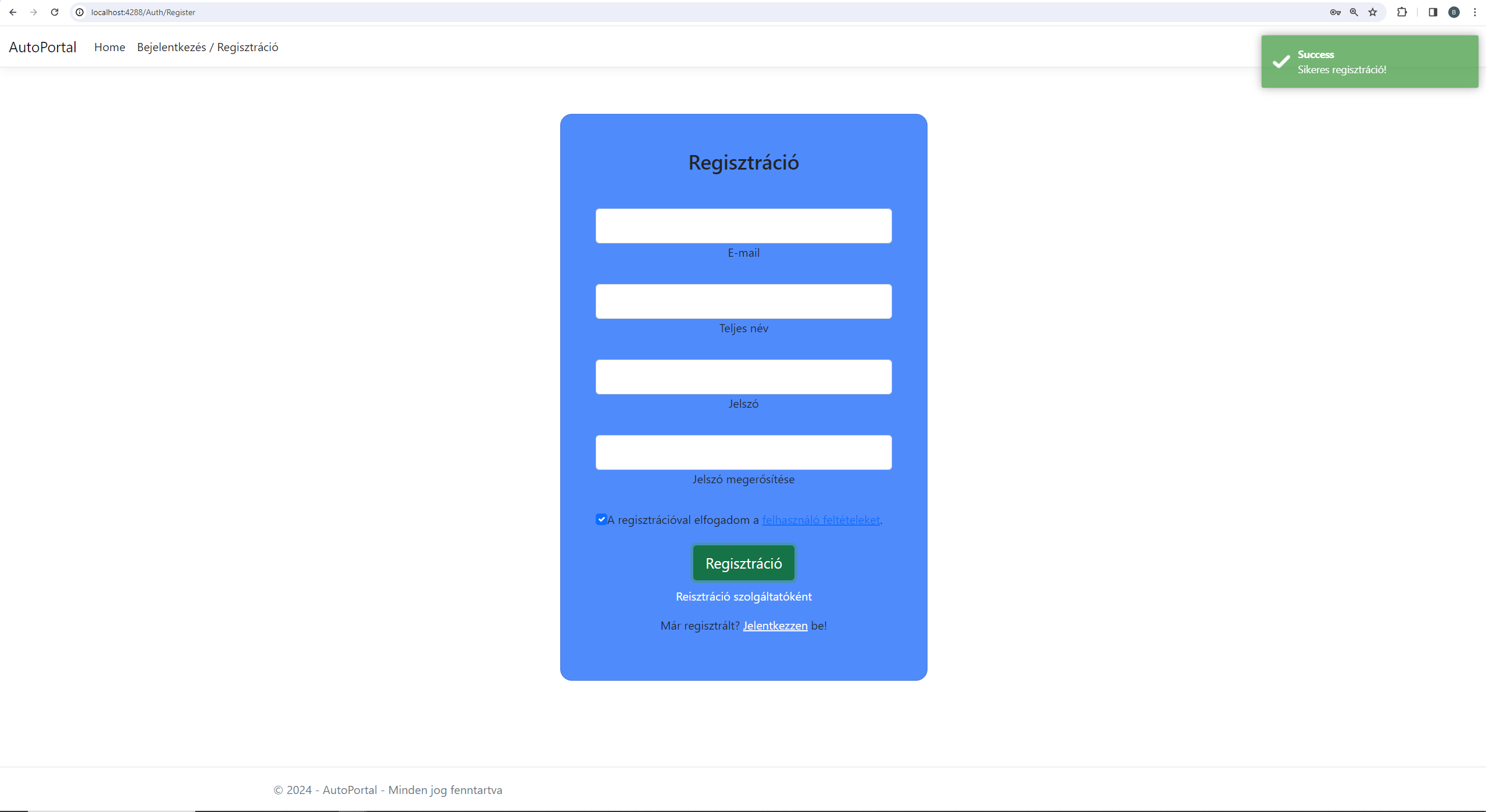


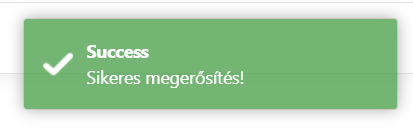
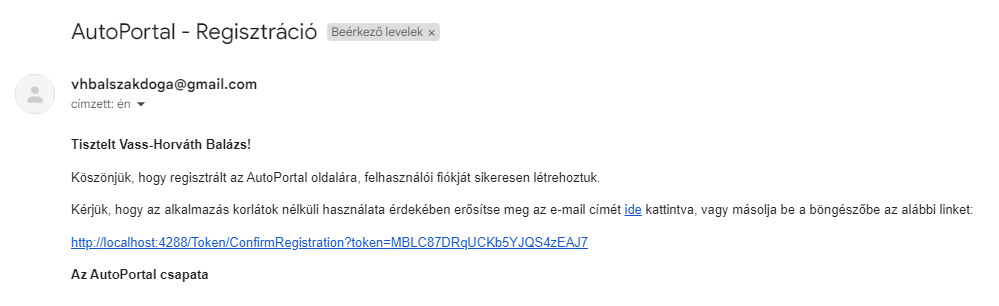
##### Gyártó

Gyártói regisztrációt közvetlenül nem támogat az oldal, ugyanis gyártói fiókot csak, és kizárólag adminisztrátor tud létrehozni biztonsági okokból. Gyártói regisztrációhoz keressük fel az oldal adminisztrátorait az elérhetőségek velemelyikén.

##### Sikeres regisztrációt követően, email cím megerősítés

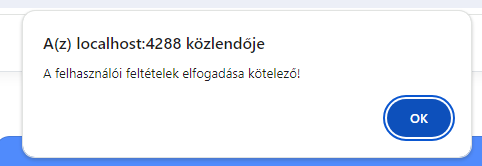
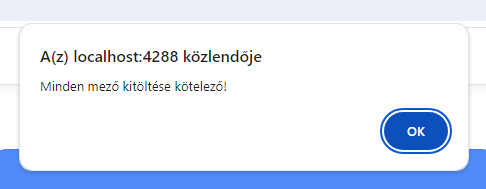
Amennyiben sikeres volt a regisztrációnk, a jobb felső sarokban egy zöld értesítés fog megjelenni. Ezt követően nemsokkal a megadott email címünkre egy megerősítő levelet fogunk kapni. Nyissuk meg az email, majd kövessük az abban található utasításokat.: Kattintsunk az „ide” hivatkozásra, vagy másoljuk be a böngészőnk címsorába a kapott linket. Ezt követően automatikusan a bejelentkezés oldalra navigál minket az oldal, valamint a jobb felső sarokban szintén megjelenik egy doboz, mely az email címünk megerősítéséről tanúskodik.



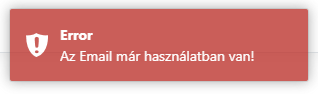
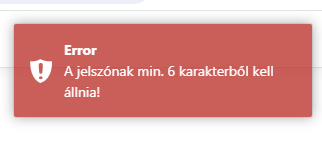


##### Lehetséges hibák regisztráció során

Az alkalmazásba történő regisztráció során belefuthatunk néhány hibába, mely a regisztráció sikertelenségét eredményezheti. Ilyen hibák lehetnek, ha az adatokat nem megfelelően, vagy hiányosan töltjük ki, vagy már regisztrált email címmel próbálunk meg újra regisztrálni. Amennyiben ilyen hibát kapnánk ellenőrizzük a megadott adatokat, és javítsuk a hibát. Az alkalmazás minden esetben egyértelmű hibaüzenetet ad, hogy tudjuk mit kell javítanunk.

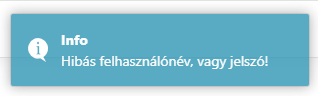
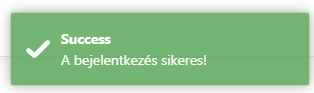


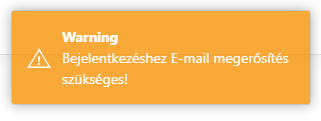
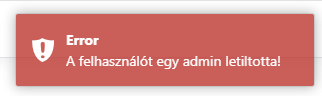




#### Bejelentkezés

Bejelentkezésre gyártó kivételével minden fióktípussal a felső menüsor bal oldalán található „Bejelentkezés / Regisztráció” gomb megnyomását követően van lehetőségünk. A megjelenő oldalon adjuk meg az email címünket, és a hozzá tartozó jelszavunkat. Miután megfelelően kitöltöttük a mezőket kattintsunk a „Bejelentkezés” gombra. Sikeres bejelentkezést követően az oldal felső menüsorában meg fognak jelenni számunkra a jogosultságainknak megfelelő menük. Hibásan kitöltött bejelentkezési mezők esetén az oldal tájékoztat minket a sikertelen bejelentkezésről. Amennyiben megszegtük az oldal felhasználói feltételeit előfordulhat, hogy egy adminisztrátor letiltja a felhasználói fiókunkat, ebben az esetben nem fogunk tudni bejelentkezni. Továbbá szolgáltatói fiók esetén a bejelentkezés feltétele az email cím megerősítése.



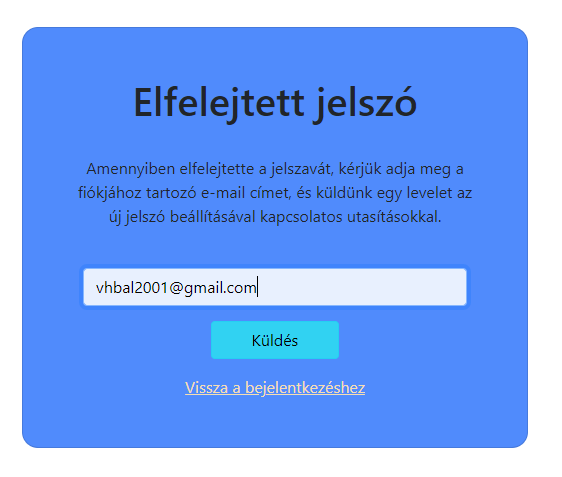
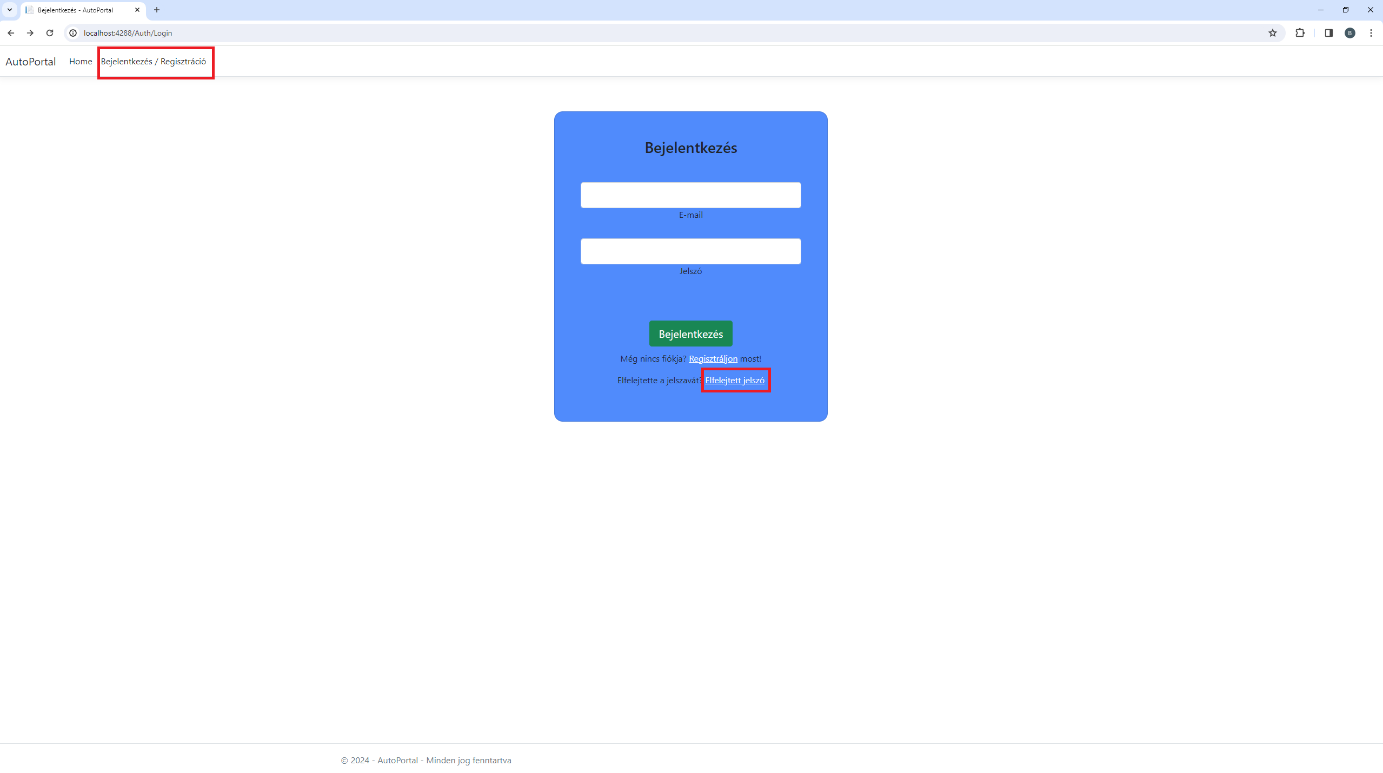


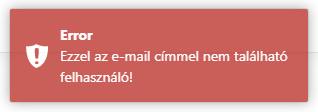
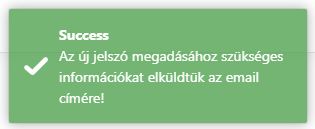
#### Elfelejtett jelszó

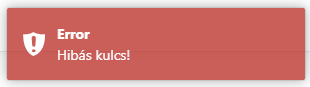
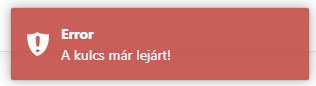
Amennyiben a felhasználó követi azt a javaslatot, miszerint érdemes egyedi, akár jelszó generátor által generált jelszót használni, ebben az esetben könnyen előfordulhat, hogy elfelejti azt. Annak érdekében, hogy ez ne történhessen meg érdemes lehet valamilyen jelszókezelő programot használni, mint például KeePass, mely elősegíti a bonyolult, nagy mennyiségű jelszavak biztonságos tárolását. Amennyiben ennek ellenére mégsem emlékeznénk a fiókunkhoz tartozó jelszóra, lehetőségünk van új jelszó beállítására, amennyiben hozzáférünk ahhoz az emailhez, amivel regisztráltunk.

Válasszuk a felső menüsorban a „Bejelentkezés / Regisztráció” menüpontot, majd kattintsunk az „Elfelejtett jelszó” hivatkozásra. A megjelenő ablak szövegdobozába írjuk be azt az email címet, amiről regisztráltunk, majd kattintsunk a „Küldés” gombra. Ekkor, ha megfelelő címet adtunk meg, a jobb felső sarokban egy sikeres üzenetet fogunk kapni, majd pillanatokon belül meg fog érkezni a levél a levelező fiókunkba. Hibás cím esetén egy hibaüzenettel tájékoztat minket a rendszer, hogy a megadott email címhez nem tartozik fiók. Nyissuk meg a kapott levelet, majd kattintsunk az „ide” hivatkozásra, vagy másoljuk be a kapott linket a böngészőnk címsorába. A megjelenő oldalon mind a két mezőbe adjuk meg az újonnan használni kívánt jelszót, majd kattintsunk a mentés gombra. Ekkor a jobb felső sarokban egy sikeres üzenetet fogunk kapni, majd nemsokára a bejelentkező oldalra navigál minket az oldal, ahol immáron az újonnan megadott jelszóval léphetünk be.

Fontos megjegyezni, hogy az emailben kapott link csak egy használatra érvényes, és legfeljebb 24 órán át él. Amennyiben kifutunk az időből, vagy újra szeretnénk elfelejtett jelszót módosítani, akkor új emailt kell kérnünk. Ezekben az esetekben „Hibás kulcs”, vagy „A kulcs már lejárt” hibaüzeneteket kapunk.

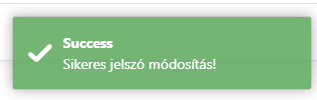






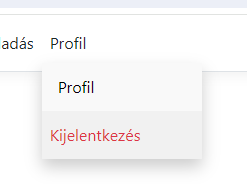






#### Kijelentkezés

Az adataink biztonsága érdekében elengedhetetlen, hogy az oldal elhagyása előtt kijelentkezzünk. Az alkalmazás a bejelentkezésünket a böngésző sütijei között tárolja, ezáltal az az oldal elhagyását követően is élni fog. A bejelentkezés „örökös” időtartamra szól, így csak akkor kell újra bejelentkeznünk, ha kijelentkeztünk. Bizonyos esetekben „elveszhet” a korábbi bejelentkezésünk, mint például böngésző adatok törlése során, vagy a böngészőn frissítésekor. Ilyen esetekben szintén újra be kell jelentkeznünk. Manuális kijelentkezésre a felső menüsorban található „Profil” menüpont alatti „Kijelentkezés” gombra kattintva van lehetőségünk. Kijelentkezést követően az alkalmazás bejelentkező oldalára fogunk kerülni.



### Fióktípusok, jogosultságok, jármű jogviszonyok

Az alkalmazásban fontos szerepet kap a felhasználói jogosultságkezelés. Az alkalmazás többszintű jogosultsági hierarchiával rendelkezik annak érdekében, hogy minél biztonságosabb, és minél jobban személyre szabható legyen. A hierarchia legfelső szintjén a bejelentkező fiók típusa van, mely lehet felhasználó, kereskedő, szerviz vagy gyártó. Bejelentkezéskor az alkalmazás az email címünk alapján automatikusan eldönti, hogy melyik kategóriába tartozunk, és ennek megfelelően határozza meg a lehetőségeinket. Felhasználó típusú fiókhoz lehetőség van további jogosultság(ok) rendelésére, mint például adminisztrátor.

A fiókok és a járművek közötti kapcsolat is többféle lehet. Egy fiókhoz korlátlan számú jármű tartozhat, valamint egy járműhöz korlátlan számú fióknak lehet jogosultsága, viszont egy járműhöz egyazon felhasználónak csak egyféle jogosultsága lehet. A jogosultsági „kapcsoló” lehet tulajdonos, üzembentartó vagy sofőr. A felhasználó a jogosultsági szintjétől függően éri el a járműhöz tartozó lehetőségeket. További jogviszony a kereskedő, mellyel csak kereskedői fióktípushoz lehet járművet kötni. Ennek célja, hogy így a járművünket a megbízott kereskedő meg tudja hirdetni eladásra.

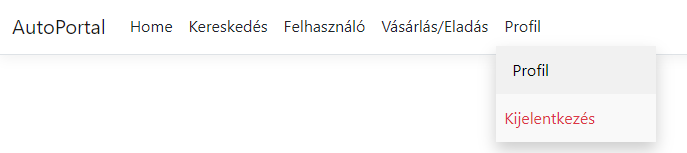
### Felső menüsor, navigálás

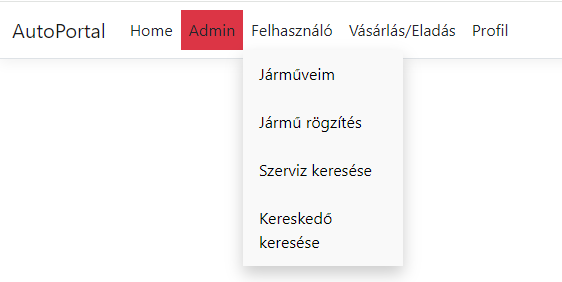
Az alkalmazásban történő navigálás a lap tetején lévő felső menüsor segítségével valósul meg. A menüsor tartalma bejelentkezésről, azt követően fiók típustól és jogosultságtól függően változik. A több funkciót tartalmazó felső menüsorok esetén azok fölé „fókuszálva” legördülő listává változnak, ahonnan kiválaszthatjuk, az almenüből, hogy mit szeretnénk csinálni.











### Általános, és felhasználói funkciók

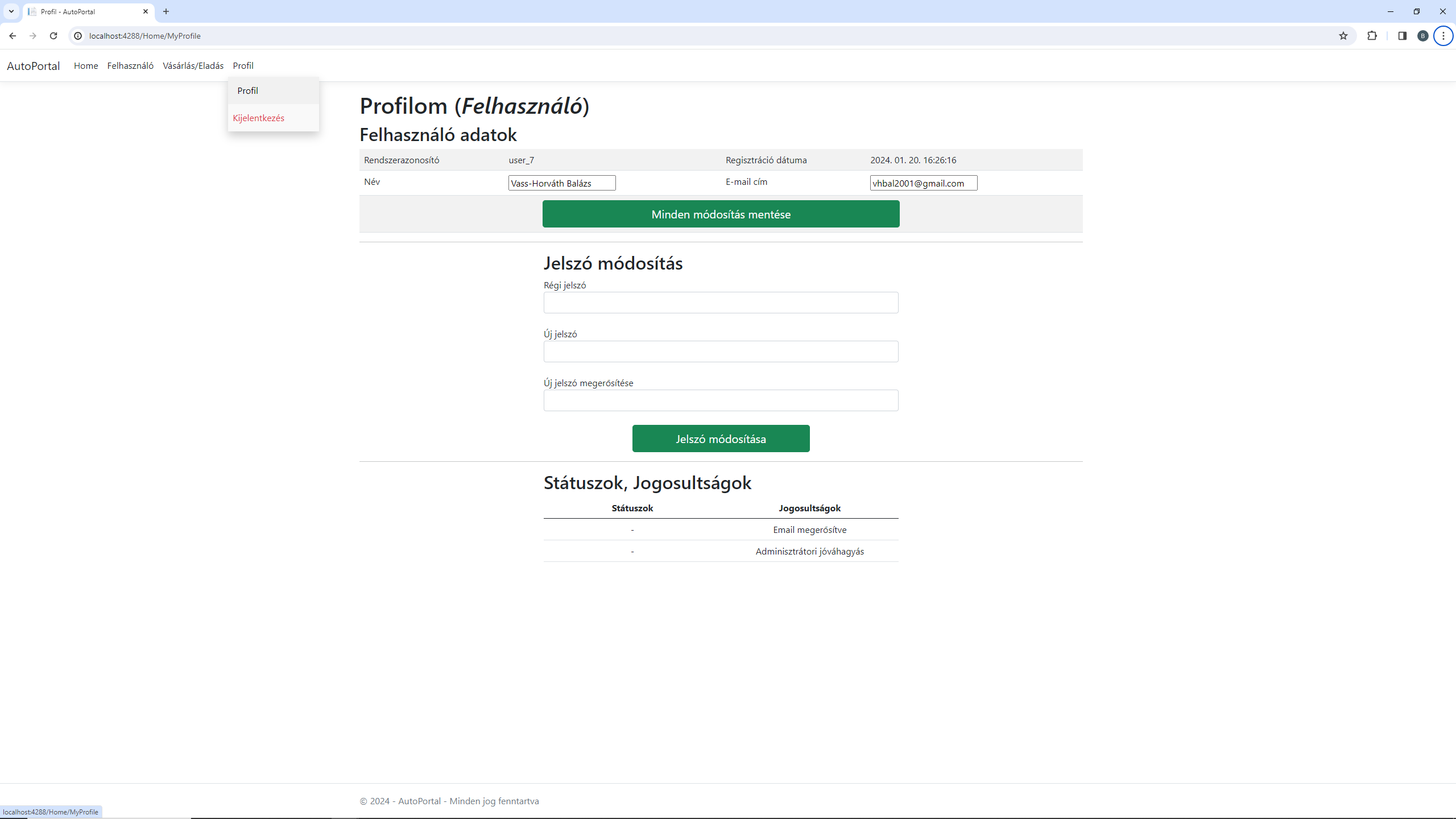
Az általános felhasználói funkciók többsége (kivétel lehet az adminisztrátori jogosultság) elérhető szolgáltató profilból is. Ennek oka, hogy egy szolgáltatónak, például szerviznek ugyan úgy lehetnek céges járművei, így ezek nyilvántartásához nem kell külön azonosítót létrehozni.

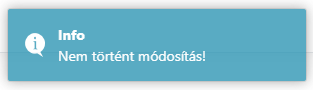
#### Profil

Amennyiben a fiókunk adatait szeretnénk megtekinteni, vagy módosítani, válasszuk a „Profil” menü „Profil” alpontját. Itt lehetőségünk van megtekinteni a rendszerazonosítónkat, a regisztrációnk dátumát.

##### Felhasználó fióktípus

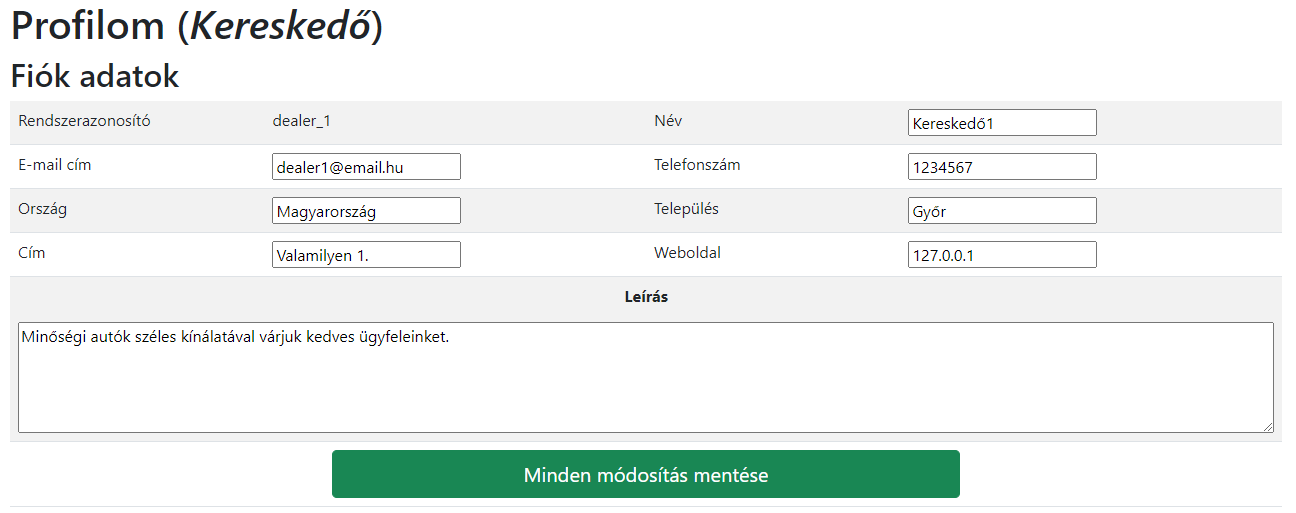
Általános felhasználói profillal lehetőségünk van módosítani a fiókunkhoz tartozó email címünket, valamint a megadott nevünket. Email cím módosításakor fontos, hogy óvatosan járjunk el. Ilyenkor is kikötés, hogy az email címnek egyedinek kell lennie, azaz nem adhatunk meg olyan címet, melyet már regisztráltunk. Ebben az esetben a rendszer tájékoztatni fog minket a hibáról. Miután az emailhez tartozó szövegdobozba beírtuk az új email címünket, és meggyőződtünk annak helyességéről, vagy módosítottuk a fiókhoz tartozó nevünket, kattintsunk a „Minden módosítás mentése” gombra. Ekkor, ha minden adatunk megfelel, a jobb felső sarokban „Sikeres módosítás” üzenet fog megjelenni. Amennyiben módosítás nélkül nyomjuk meg a gombot, úgy természetesen a rendszerben nem fog történni módosítás, és erről a felhasználó tájékoztatást is kap.





##### Szolgáltató fióktípus

Szolgáltatói profil esetén lehetőségünk van további adatok beállítására, ezzel megkönnyítve leendő ügyfeleinknek, hogy megtaláljanak minket. A regisztrációkor ezek az adatok még nem állíthatók be, így az első bejelentkezést követően érdemes megadnunk. Az elérhetőségeink között beállíthatunk többek között telefonszámot, címet (Ország, város, cím), weboldalt, valamint leírást is adhatunk az oldalunkhoz, ahol feltűntethetjük például a pontos szolgáltatásainkat, nyitvatartásunkat, vagy bármilyen egyéb fontos adatot. Miután elvégeztük a kívánt módosításokat, kattintsunk a „Minden módosítás mentése” gombra, hogy érvénybe léptessük őket.

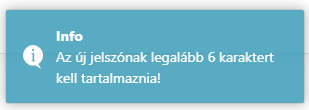
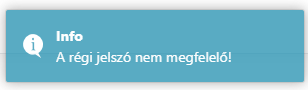


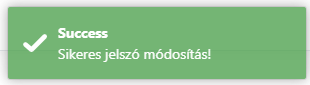
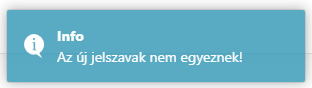
##### Jelszó módosítás

Adataink, és fiókunk biztonsága érdekében javasolt időnként módosítani a felhasználónkhoz tartozó jelszót. A Profil menüpont „Jelszó módosítása” része pont erre biztosít lehetőséget. A jelszó módosításához szükségünk van az aktuális jelszavunk megadására is, továbbá biztonsági célból a következő két mezőbe meg kell adnunk az új jelszavunkat, majd megismételni azt. Miután megfelelően kitöltöttük az aktuális, és az új jelszóra vonatkozó bemeneti mezőket kattintsunk a „Jelszó módosítása” gombra. A módosított jelszóra vonatkozó javaslatok és előírások megfelelnek a regisztrációnál leírtakkal. A módosítás eredményéről a felhasználó a jobb felső sarokban megjelenő értesítésből tájékozódhat, ezek az alábbiak lehetnek:

1. A régi jelszó nem megfelelő: Az aktuális jelszót helytelenül adtuk meg
2. Az új jelszónak legalább 6 karaktert kell tartalmaznia: Az új jelszavunk garantáltan nem biztonságos, ugyanis nem tartalmaz még 6 karaktert se. Adjuk meg legalább 6 karakter hosszúságú jelszót
3. Az új jelszavak nem egyeznek: Az új jelszó, és annak megerősítése nem ugyan azok. Ellenőrizzük, hogy a jelszó, és annak megerősítése is azonos értéket tartalmazzon.
4. Sikeres jelszó módosítás: A jelszó sikeresen módosítva lett, a következő bejelentkezéskor már azzal tudunk belépni.



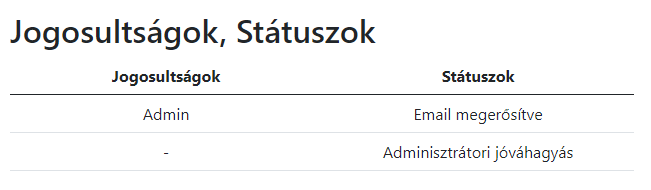
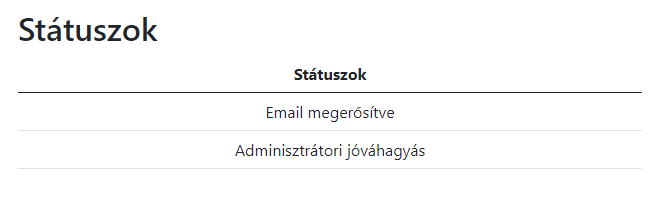




##### Státuszok, jogosultságok

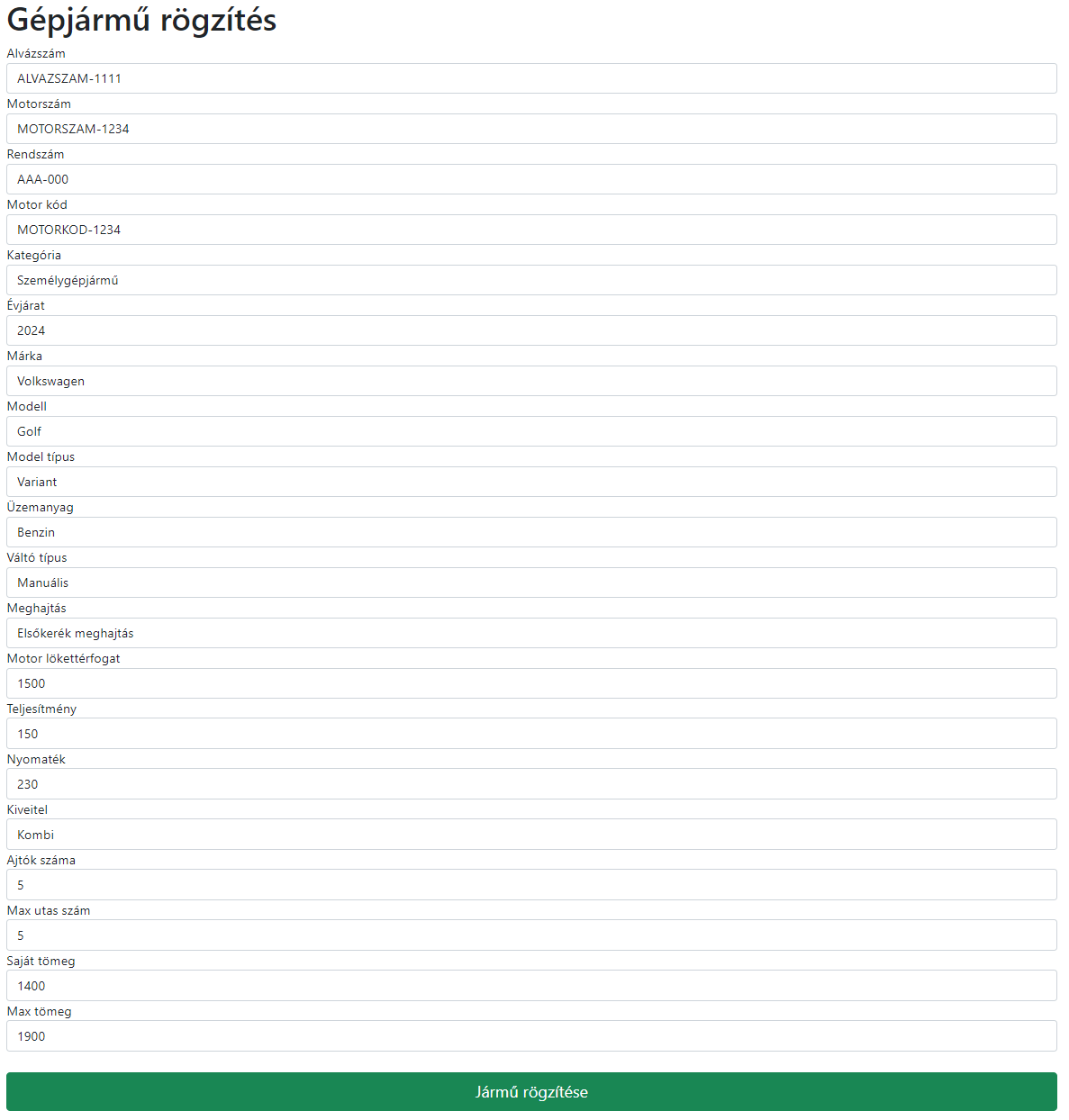
Minden fiókhoz tartozhatnak különféle státuszok, ezeket a Profil menüpont alatt ellenőrizhetjük. Státusz lehet például a megerősített email cím vagy az adminisztrátori jóváhagyás.

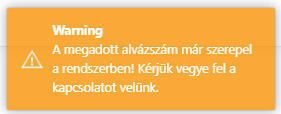
Felhasználói fióktípushoz továbbá jogosultság is rendelhető, mint például az Admin.

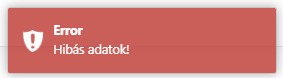


#### Jármű rögzítése a rendszerbe

Amennyiben újonnan szeretnénk gépjárművet rögzíteni a rendszerbe, navigáljunk a „Felhasználó” menüpont alatti „Jármű rögzítése” pontba. Felhasználói fiók típusától függetlenül korlátlan számban rögzíthetünk járműveket a rendszerbe. Egy járművet csak egyszer lehet rögzíteni, a járműveket az alvázszám alapján különböztetjük meg egymástól. Amennyiben most vásároltunk használtan egy járművet, és azt a korábbi tulajdonosa már regisztrálta a rendszerben, úgy már nincs szükségünk újbóli rögzítésre, szimplán a korábbi tulajnak kell a fiókunkra átruháznia a tulajdonosi jogosultságot. Amennyiben erre nincs lehetősége, keressünk fel egy adminisztrátor. Jármű rögzítésekor a megjelenő mezők mindegyikét a megfelelő adatokkal kell kitölteni. Meg kell adnunk többek között a jármű jellegét, kivitelét, márkáját, típusát, évjáratát, alvázszámát, különböző hajtáslánc, és egyéb műszaki adatokat. Márka esetén segítségünkre van egy legördülő lista, viszont magunk is beírhatjuk a jármű gyártóját, ha nem a listában még nem szerepel. Amennyiben elkezdünk adatokat beírni a márkához tartozó szövegdobozba, úgy az folyamatosan szűrni fogja a megjelenített találatokat. Az üzemanyag típusánál, váltó típusnál, valamint meghajtásnál csak a legördülő listában megjelenő adatok valamelyikét választhatjuk. A lökettérfogat, teljesítmény, nyomaték, ajtók száma, max utasszám, saját tömeg, max tömeg mezőkbe csak és kizárólag egész számokat adhatunk meg, a kivitel mezőben a márkához hasonlóan lehetőségünk van listából választani, vagy saját adatot beírni. Miután minden adatot megfelelően kitöltöttünk, kattintsunk a „Jármű rögzítése” gombra. Amennyiben minden adatot sikeresen rögzítettünk, a bal felső sarokban egy értesítést fogunk kapni, miszerint a járművet sikerült rögzíteni. Amennyiben a jármű már szerepel a rendszerben, egy figyelmeztető, hibás, és hiányos adatok esetében pedig hibaüzenetet kapunk. A jármű rögzítése után az oldalon található mezők nem vesztik el a tartalmukat, ennek célja, hogy amennyiben egy flottakezelő cég rögzít több azonos járművet a rendszerbe, úgy ne kelljen azok adatait mindig újra egyesével kitölteni, szimplán elég a különböző mezők (például alvázszám, rendszám) értékeinek lecserélése, míg a márka, modell, és többi egyező adatunk megmarad.

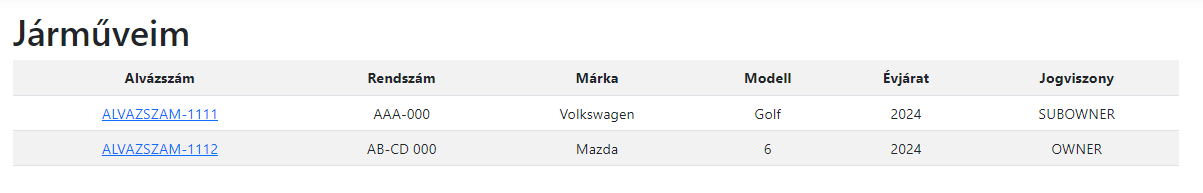






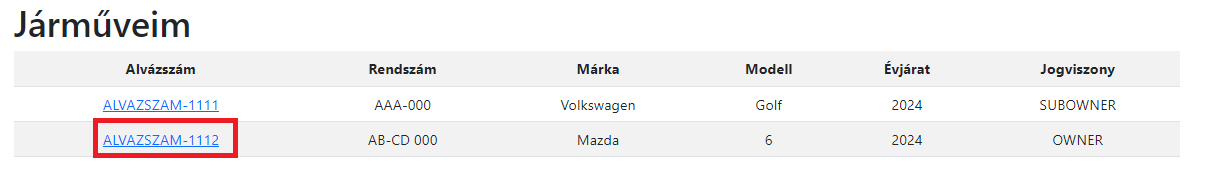
#### Saját járművek áttekintése

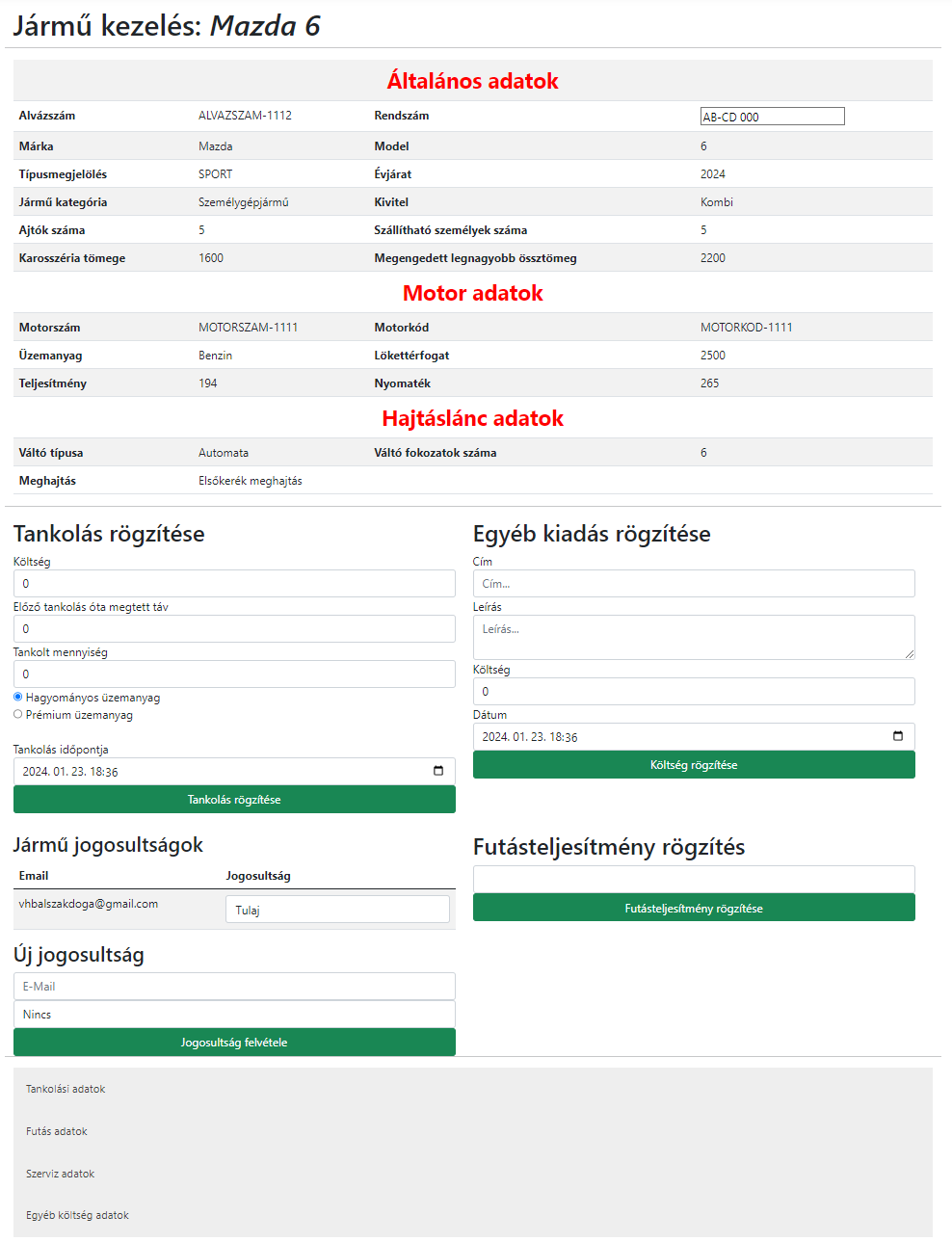
Amennyiben szeretnénk áttekinteni a saját, vagy a hozzánk kapcsolódó járműveket válasszuk a „Felhasználó” menüpont alatti „Járműveim” opciót. Ekkora megjelenő oldalon megjelenik az összes olyan jármű, melyeknek a tulajdonosa vagyunk, vagy valamilyen jogosultságunk van a járműhöz. Az előnézetben látjuk a járművek alvázszámát, rendszámát, márkáját, modelljét, évjáratát, valamint azt, hogy milyen jogviszony köti hozzánk őket.



#### Járműrészletező, járműkezelés

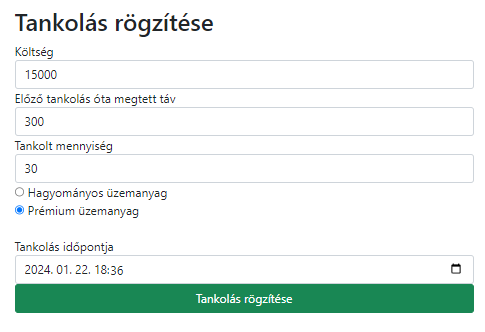
Amennyiben szeretnénk megtekinteni valamelyik járművünkhöz kapcsolódó adatokat, vagy kezelni azokat, abban az esetben a járműáttekintőben kattintsunk a kívánt jármű (kéken megjelenő) alvázszámára. Ekkor az alkalmazás elnavigál minket a járműkezelő oldalra. Az itt megjelenő adatok, és lehetőségek a jármű és felhasználó között lévő jogviszony függvényében változnak. Megtekinthetjük a járművünk minden általános, műszaki, és hajtáslánc adatát, kiadásokat rögzíthetünk, jogosultságot kezelhetünk, valamint ellenőrizhetjük az előéletét a korábbi adatok alapján.





##### Tankolás rögzítés

A járművünkhöz már sofőr jogosultsággal is lehetőségünk van tankolást rögzíteni. Ekkor meg kell adnunk a költséget, az előző tankolás óta megtett távot, továbbá a tankolt üzemanyag mennyiségét és típusát (hagyományos, prémium). A tankolás dátuma alapértelmezetten a mai nap, viszont ezt a dátumválasztóban módosíthatjuk. Miután kitöltöttük a kívánt adatokat kattintsunk a „Tankolás rögzítése” gombra. Ekkor a jobb felső sarokban egy értesítést fogunk kapni, miszerint „A tankolás sikeresen rögzítve” lett. A rendszer automatikusan számol nekünk 100km-re vetett átlagfogyasztást, valamint átlagköltséget.



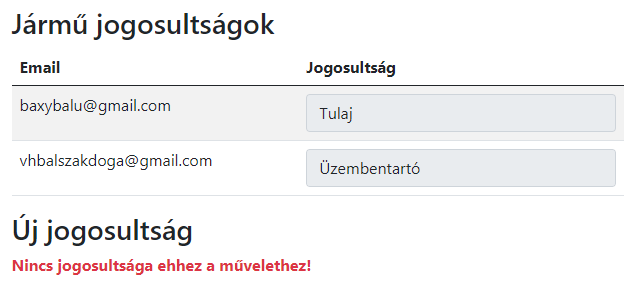
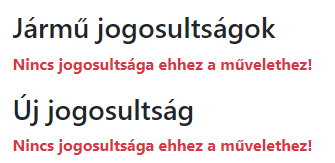
##### Egyéb kiadás rögzítése

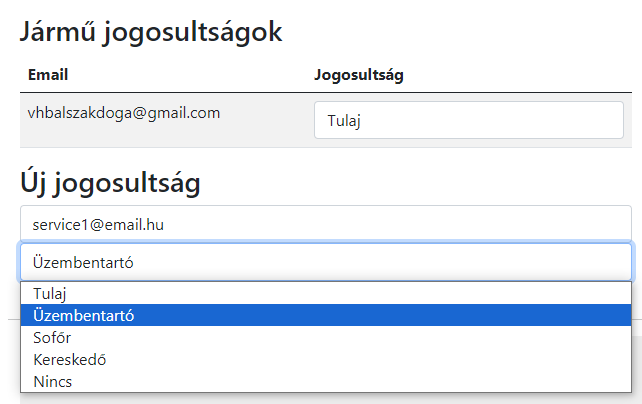
A járművünkhöz lehetőségünk van egyéb kiadás(ok) rögzítésére. Ide tartozhatnak például a kötelező biztosításból, casco biztosításból, tartozékvásárlásból, autómosásból, kerékcseréből, autóápolószer vásárlásból, vagy bármilyen rendkívüli okból adódó költségek, kiadások. Egyéb kiadás rögzítésére sofőr jogosultsággal nincs lehetőség, ezt csak tulajdonos, és üzembentartó teheti meg. A kiadáshoz meg kell adnunk egy címet, leírást, költséget, és a kiadás dátumát. Miután minden adatot megfelelően kitöltöttünk kattintsunk a „Költség rögzítése” gombra. Ekkor a jobb felső sarokban meg fog jelenni egy sikeres műveletre utaló értesítés.



##### Jogviszonykezelés

A fiók- és a jármű közötti jogviszony kezelése két részre van bontva. Üzembentartóként láthatjuk, hogy kiknek milyen jogosultsága van a szóban forgó járműhöz, mindazonáltal módosítani nem tudunk rajta. Tulajdonosként lehetőségünk van a járműhöz tartozó jogosultságok módosítására. Amennyiben egy felhasználónak már van jogosultsága a járművünkhöz, és azon szeretnénk változtatni (például sofőr helyett üzembentartó), vagy elvenni azt, úgy a táblázatban az email cím melletti legördülő listában válasszuk ki, hogy mi legyen a felhasználó új jogviszonya. Amennyiben újonnan szeretnénk valakinek jogosultságot adni a járművünkhöz, úgy az „Új jogosultság” alatt adjuk meg a felhasználó email címét, a legördülő listából pedig válasszuk ki a kívánt jogosultságot, majd kattintsunk a Jogosultság felvétele gombra. Ekkor sikeres értesítést fogunk kapni a jobb felső sarokban, valamint egyből meg fog jelenni a táblázatban az új felhasználó, és a járműhöz tartozó jogosultsága.



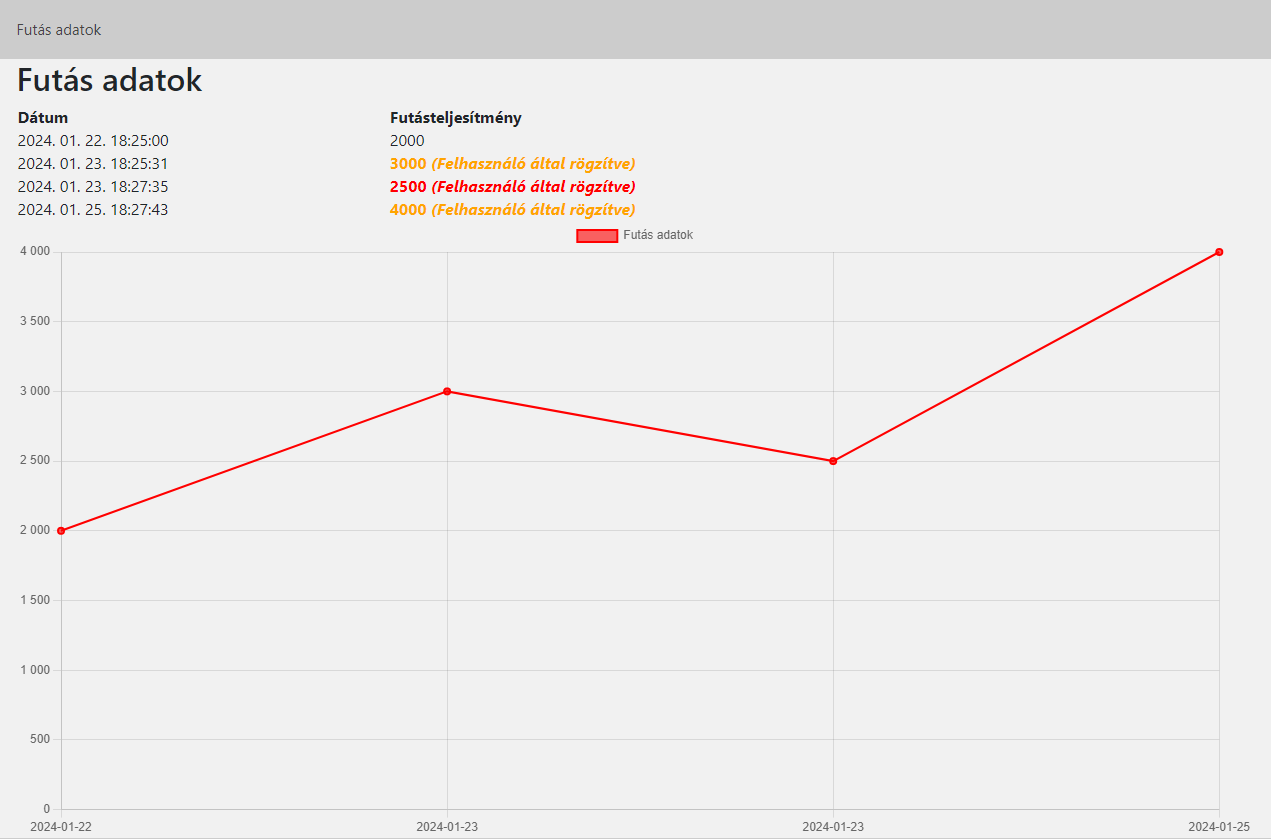


##### Járműhöz rögzített információk

A járművünkhöz rögzített tankolási adatokat, futásteljesítmény adatokat, szerviz adatokat, valamint egyéb költségeket a lap alján megjelenő lenyitható részek alatt találjuk meg. A kívánt menüpont lenyitásához kattintsunk a menüpont címsorába tetszőleges helyen. Ekkor amennyiben tartozik a járműhöz az adott típusú feljegyzés, akkor azok meg fognak jelenni, ellenkező esetben egy értesítést fogunk látni, hogy még nem található adat.

A futásteljesítménynél továbbá megkülönböztetünk szerviz által rögzített adatokat, valamint felhasználó által rögzített adatokat. A felhasználó által rögzített adatok sárgán jelennek meg a listában, valamint zárójelben megjelenik a „Felhasználó által rögzítve” felirat, ezzel jelezve, hogy az adat nem hiteles. Továbbá azon adatok, ahol idegenkezűség feltételezhető, mint például az óraállás visszatekerése, abban az esetben az adat pirosan jelenik meg. A jobb átláthatóságot egy vizuálisan megjelenő grafikon segíti, melynek Y tengelye az óraállás, az X tengely pedig a dátum. Egy naphoz több futási adat is tartozhat, ebben az esetben az X tengelyen többször is megjelenhet egyazon nap, idő (óra-perc-másodperc) szerint rendezve. A grafikon pontjára fókuszálva az egérrel továbbá láthatjuk a rögzítés dátumát, és a pontos értéket.

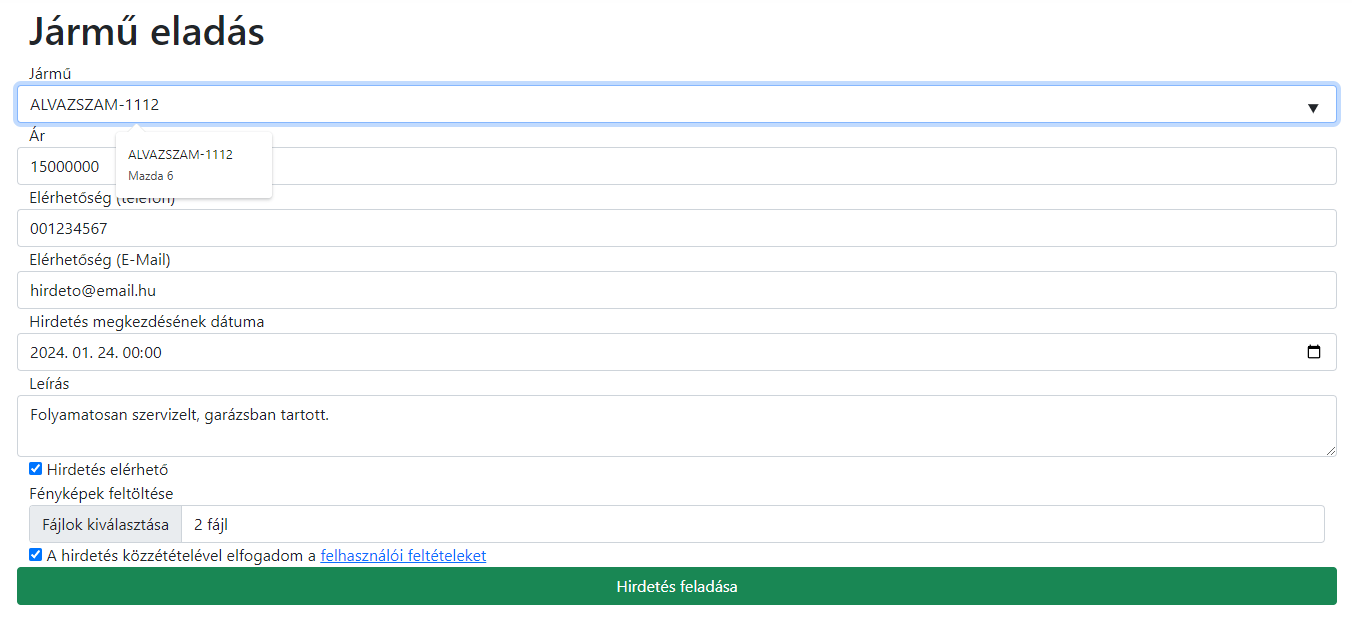


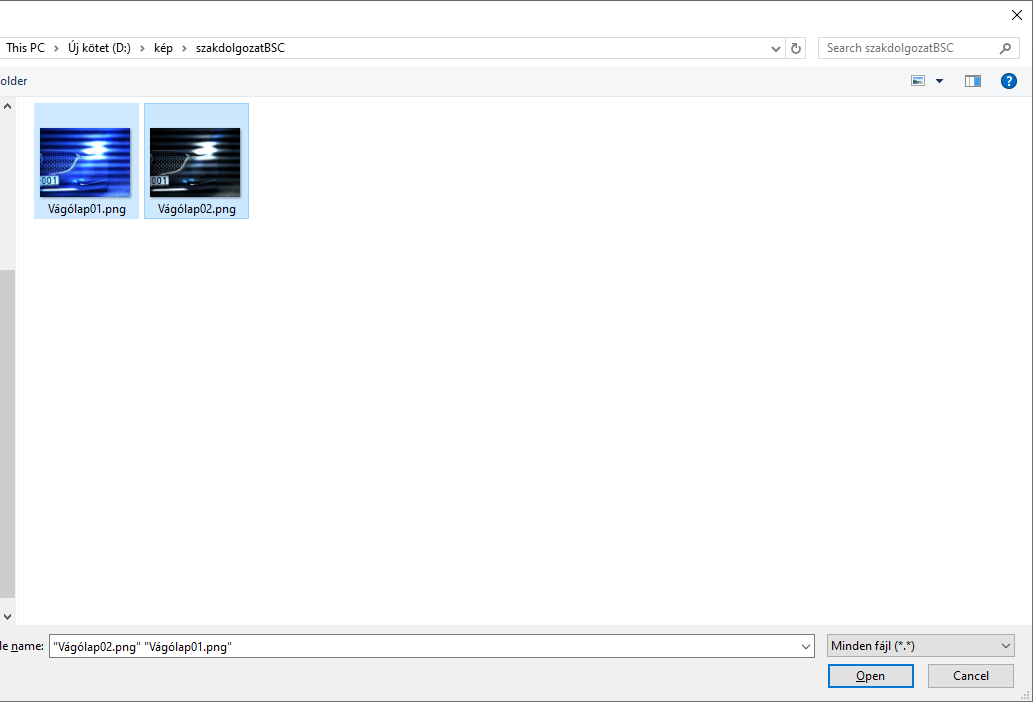


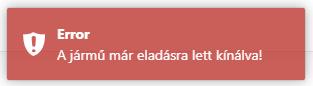
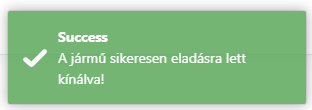


#### Jármű eladásra kínálása

Az alkalmazás elősegíti a gépjármű adás-vétel folyamatát is. Felhasználói fiók típusától függetlenül lehetőségünk van járműveink eladásra kínálására. Kikötés, hogy nem kereskedő fióktípussal legfeljebb 2 járművet hirdethetünk eladásra egyidejűleg. Jármű eladásához válasszuk a „Vásárlás/Eladás” menüpont alatt található „Jármű eladása” opciót. Fontos kikötés, hogy járművet csak, és kizárólag tulajdonosi (vagy kereskedő esetén kereskedői) jogviszonnyal lehet eladásra kínálni. A megjelenő oldalon töltsük ki a szükséges mezőket. A jármű mezőbe a jármű alvázszámát kell megadnunk, melyet listából is kiválaszthatunk, vagy ha elkezdjük beírni, akkor folyamatosan szűrésre kerülnek a találatok. A meghirdetett járműhöz kötelező megadni árat, valamint telefonos, és emailes elérhetőséget. Amennyiben nem azonnal szeretnénk eladásra kínálni a járművünket lehetőségünk nyílik beállítani egy kezdő dátumot, amikortól a hirdetés megjelenik a felhasználók számára, valamint a „Hirdetés elérhető” dobozzal manuálisan kapcsolhatjuk ki-be a hirdetésünket. Adhatunk meg leírást, valamint tölthetünk fel fényképeket a járművünkről. A rendszer csak, és kizárólag .png, .jpg, vagy .jpeg kiterjesztésű képeket fogad el, bármilyen más kiterjesztésű fájl automatikusan eldobásra kerül. Hirdetés közzétételhez kötelezően el kell fogadnunk a felhasználói feltételeket. Miután kidolgoztuk a hirdetésünket, kattintsunk a „Hirdetés feladása” gombra. Amennyiben mindent jól csináltunk, a jobb felső sarokban egy értesítést fogunk kapni, miszerint a hirdetésünk bekerült a rendszerbe. Egy jármű egyidőben csak egy hirdetésben szerepelhet. Amennyiben már közzé van téve a hirdetésünk, hibaüzenetet fogunk kapni.







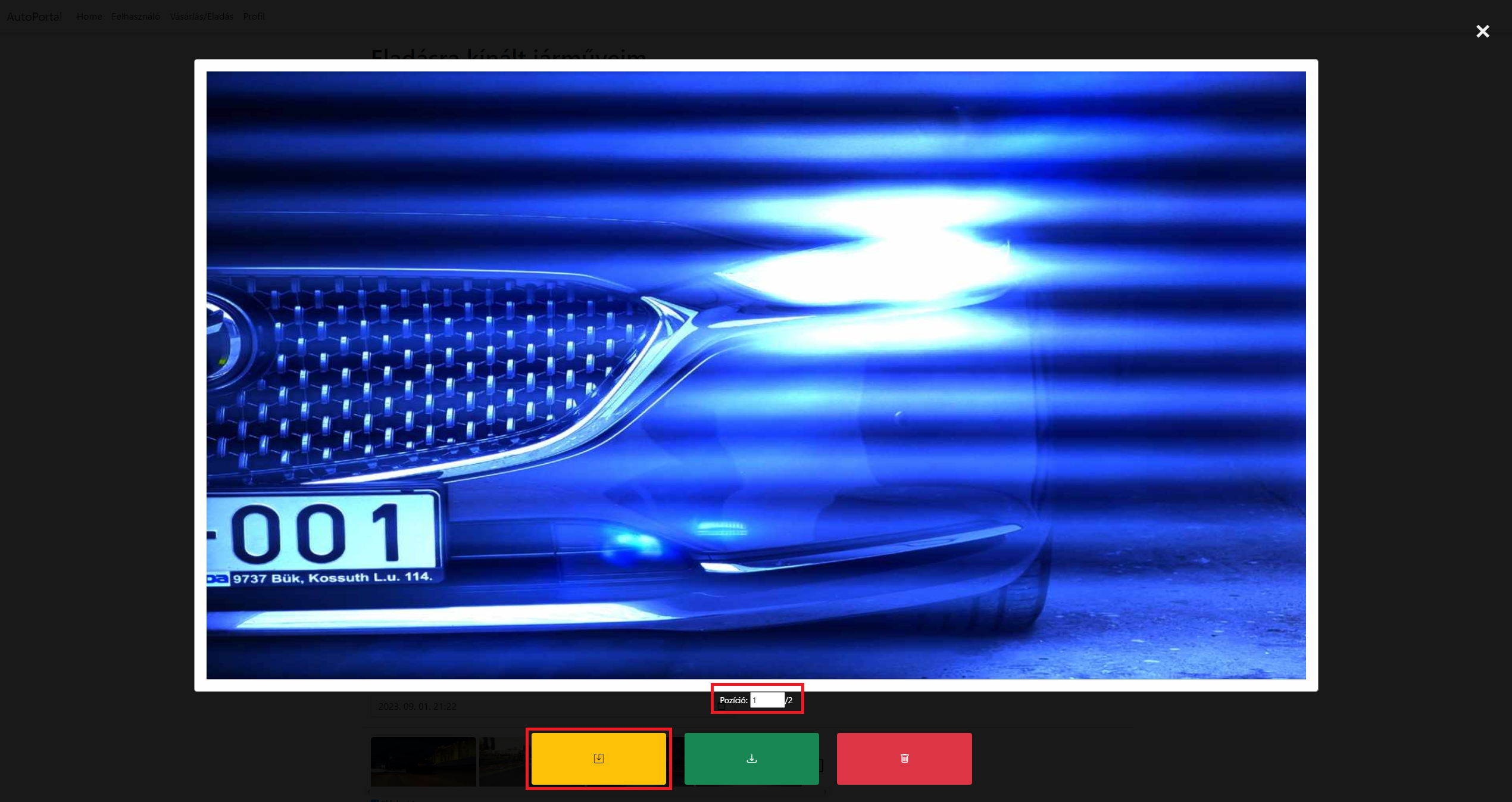
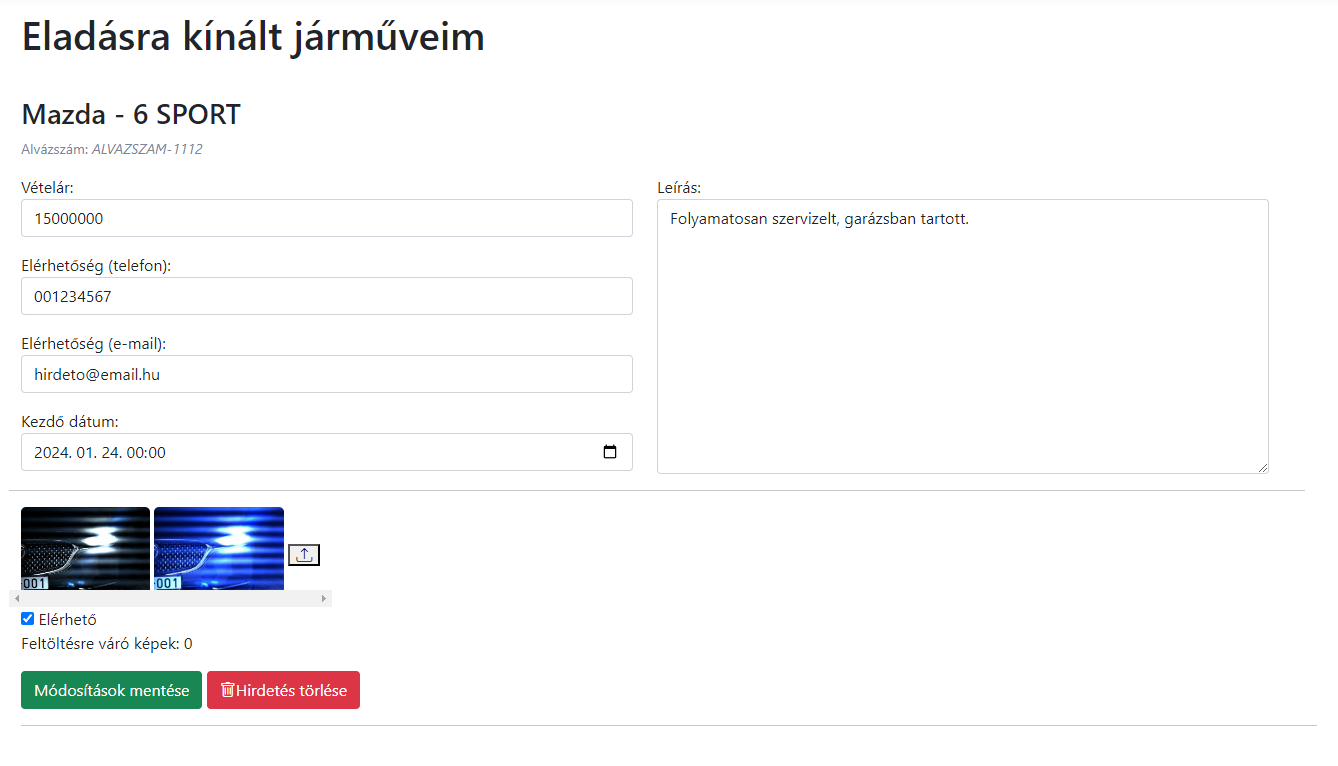
#### Feladott hirdetés módosítása, törlése

Amennyiben egy már közzétett hirdetésünket szeretnénk módosítani, vagy törölni, akkor válasszuk a „Vásárlás/Eladás” menüpont alatti „Eladó járműveim” opciót, itt meg fog jelenni az összes aktuális hirdetésünk. A hirdetés bármely adatát módosíthatjuk, a jármű (alvázszáma) kivételével. Lehetőségünk van továbbá a hirdetés közzétételének időpontját is módosítani, vagy ki-be kapcsolni a hirdetést az „Elérhető” opció ki, vagy bepipálásával. További kép(ek)et tölthetünk fel, vagy törölhetünk a jelenlegi képek közül, de a képek sorrendjén is módosíthatunk. A hirdetéskeresőben mindig az első kép lesz a hirdetés kezdőképe, és csak az fog megjelenni az előnézeti módban, a többi kép csak a részletező oldalon fog látszani.

Fontos, hogy a módosítások érvényesítéséhez mindig ne felejtsünk el a „Módosítások mentése” gombra kattintani.

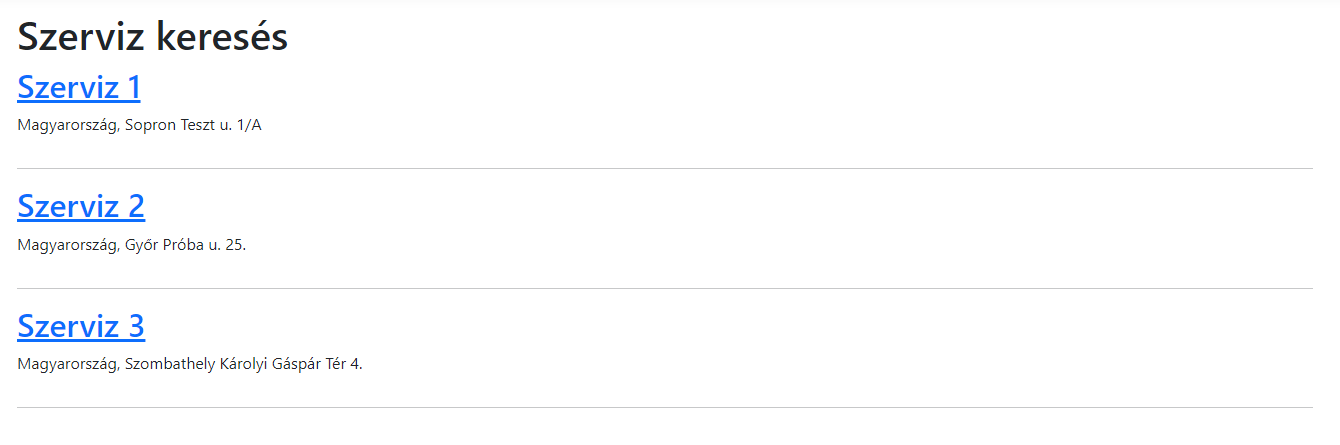
Kép törléséhez kattintsunk a kép listából a megfelelő képre, ekkor megjelenik a kép nagyobb nézetben. A törléshez válasszuk az alul megjelenő gombok közül a piros, fehér kuka piktogramot tartalmazó gombot. Továbbá lehetőségünk van innen letölteni a képet a zöld letöltés ábrás gombbal. A képek sorrendjének a módosításához a kép alatt található „Pozíció” mezőbe írjuk be, hogy hányadik helyen szeretnénk a képet megjeleníteni, majd kattintsunk a sárga gombra. Ilyenkor az adott pozíción lévő képpel összecseréljük őket, ezt követően egy „Kép sikeresen mozgatva” értesítés fog megjelenni, majd életbe lép a módosítás.

Amennyiben nem módosítani, hanem törölni szeretnénk a hirdetést, abban az esetben kattintsunk a „Hirdetés törlése” feliratú piros gombra.



#### Szerviz / Kereskedő keresése

Az alkalmazásban a szolgáltatói profiloknak a többlet funkciókon túli szerepe, hogy a felhasználóknak megkönnyítsük a megtalálásukat. Lehetőségünk van szerviz, vagy kereskedő keresésére, ehhez válasszuk ki a „Felhasználó” menüpont alatt található „Szerviz keresése”, vagy „Kereskedő keresése” opciót. A megjelenő oldalon láthatjuk az alkalmazásban regisztrált összes elérhető szolgáltatót. Az inaktív, és kitiltott szolgáltatók természetesen nem kerülnek be a listába.

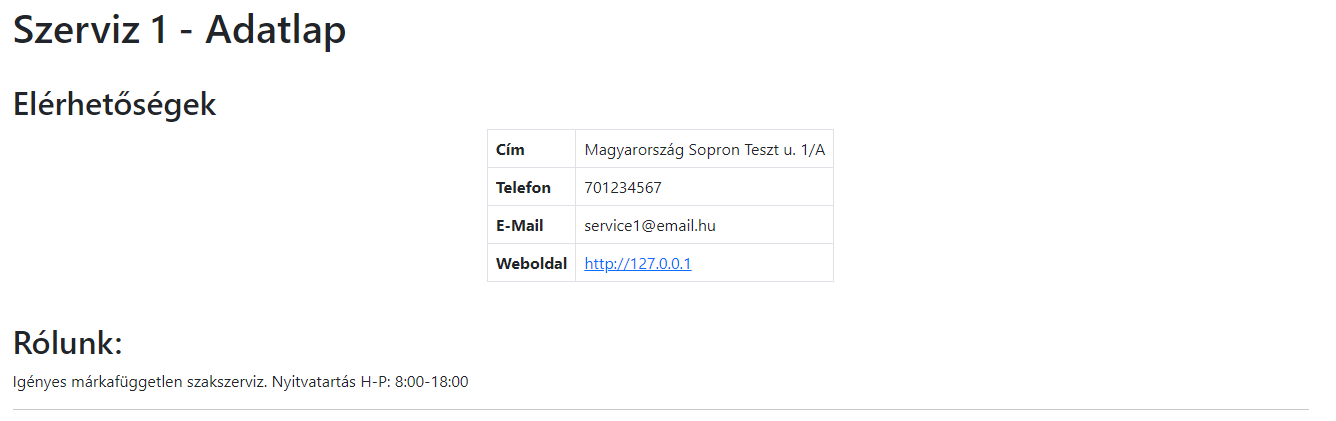


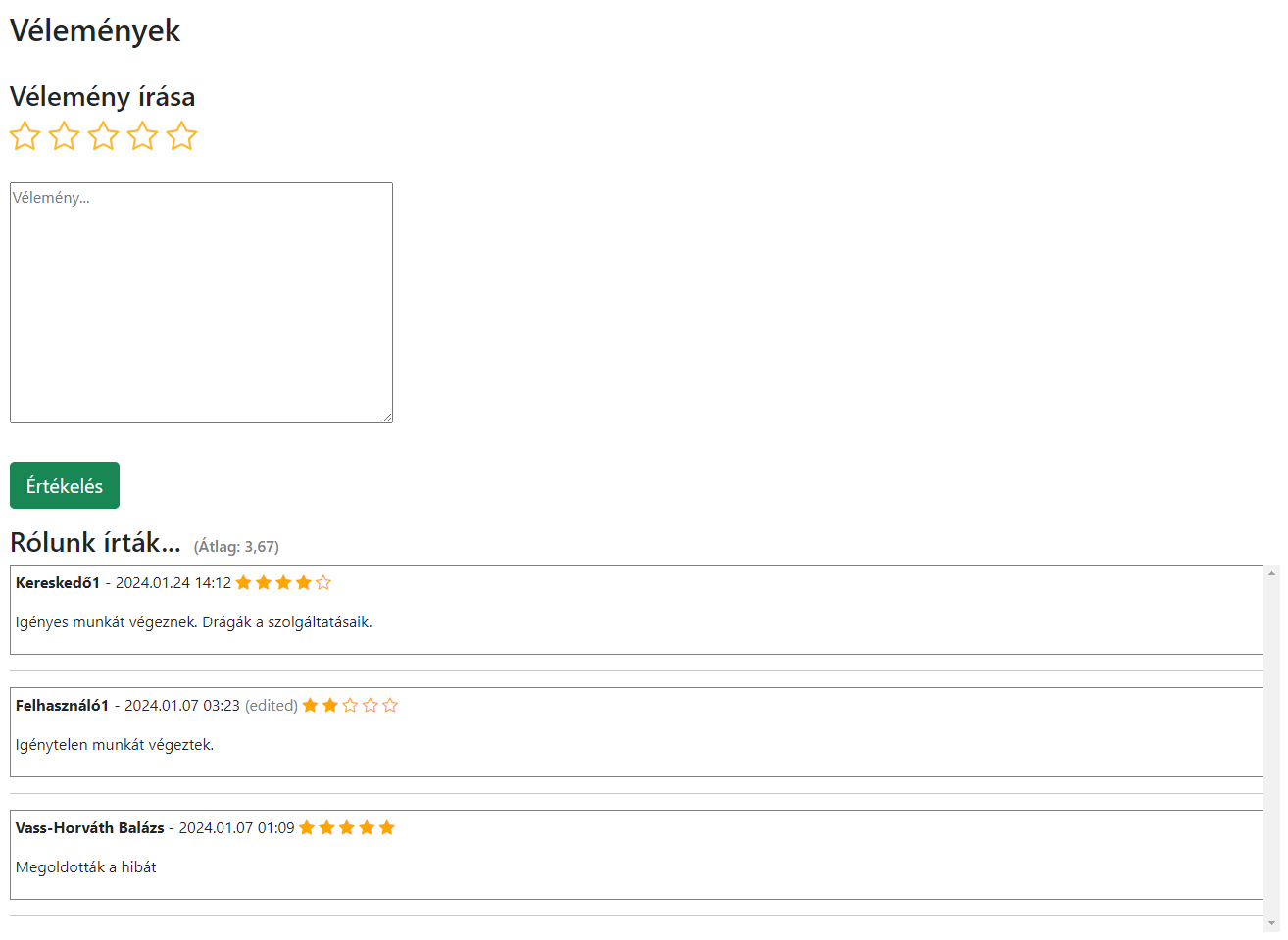
#### Szolgáltató publikus profil, szolgáltató értékelése

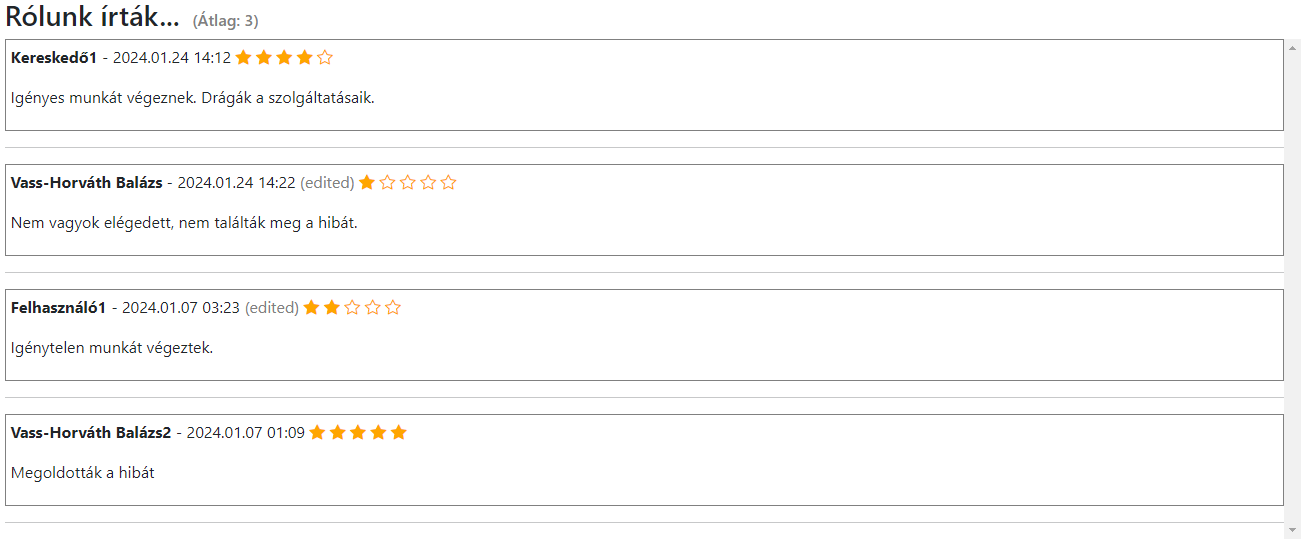
A szolgáltató publikus profiljának megtekintéséhez kattintsunk a szolgáltatókeresőben a szolgáltató nevére (kékkel kiemelve).

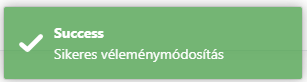
Általában mielőtt igénybe veszünk egy költséges szolgáltatást szeretünk utánajárni, hogy az adott szolgáltatóról milyen véleményeket írtak mások, valamint mi is megoszthassuk saját tapasztalatainkat. A szolgáltató publikus profilján többek között megtekinthetjük a szolgáltató adatait, és elérhetőségeit, valamint megtekinthetjük azokat a véleményeket, amiket róluk írtak, ha van olyan. Ezen túl magunk is véleményezhetjük a szolgáltatót, ezt egy 1-től 5-ig terjedő skálával, valamint egy szöveges értékelési lehetőséggel segíti elő az alkalmazás. A véleményeknél láthatjuk az összesített átlagot 2 tizedesjegyre kerekítve, valamint a felhasználók által írt véleményt, a megírás dátumát, valamint a kapott csillagok számát. Amennyiben a véleményt módosította a felhasználó, abban az esetben zárójelben megjelenik az „edited” felirat.

Amennyiben magunk is szeretnénk véleményt írni a szolgáltatóról, nincs más dolgunk, mint a publikus profilban a „Vélemény írása” alatt kiválasztani hány csillagot szeretnénk adni 1-től 5-ig, valamint a vélemény szövegdobozba részletesen kibontani a véleményünket, tapasztalatainkat, majd az „Értékelés” gombra kattintani. Amennyiben már írtunk véleményt a szolgáltatóról, abban az esetben automatikusan ki lesz töltve a vélemény írása a korábbi értékelésünkkel, ha az oldalra navigálunk. Amennyiben korábbi véleményünket módosítani szeretnénk, csak írjuk át a szöveges értékelést, valamint válasszuk ki újból, hány csillagot szeretnénk adni, majd kattintsunk ismét az értékelésre. Vélemény törléséhez egyszerűen csak kattintsunk a piros színű „Értékelés törlése” gombra. Egy felhasználó egy szolgáltatóhoz csak egy véleményt írhat, szükség esetén azt módosíthatja, vagy törlés után újra írhatja.





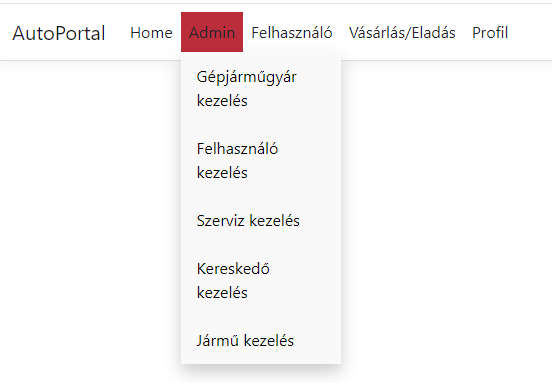
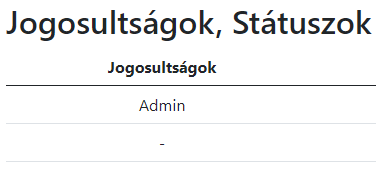




### Adminisztrátor specifikus funkciók

Az alkalmazásban rendkívüli szerepet kap az adminisztrátori jogkör, ugyanis gyakorlatilag minden adatot módosíthat koráltok nélkül. Az adminisztrátoroknál továbbá fontos a felelősségteljesség, és odafigyelés, ugyanis az alkalmazás az adminisztrátori műveletek során csupán olyan ellenőrzéseket végez, és azon műveleteket tiltja csak le, amik az alkalmazás működésében kárt tehetnek. Azaz nem ellenőrizzük például egy felhasználó email címének módosításakor, hogy valós lehet-e a megadott cím, viszont meggyőződünk róla, hogy a rendszerben még nem szerepel.

Adminisztrátori jogkör felhasználói fiókhoz kapcsolódhat. Amennyiben a felhasználói rendelkezik a jogkörrel, abban az esetben a felső menüsorban megjelenik az „Admin” menü, kiemelve piros háttérrel. Továbbá a profilunk alatt a jogosultságoknál is megjelenik, hogy adminisztrátori jogkörünk van.

#### Gépjárműgyár kezelés

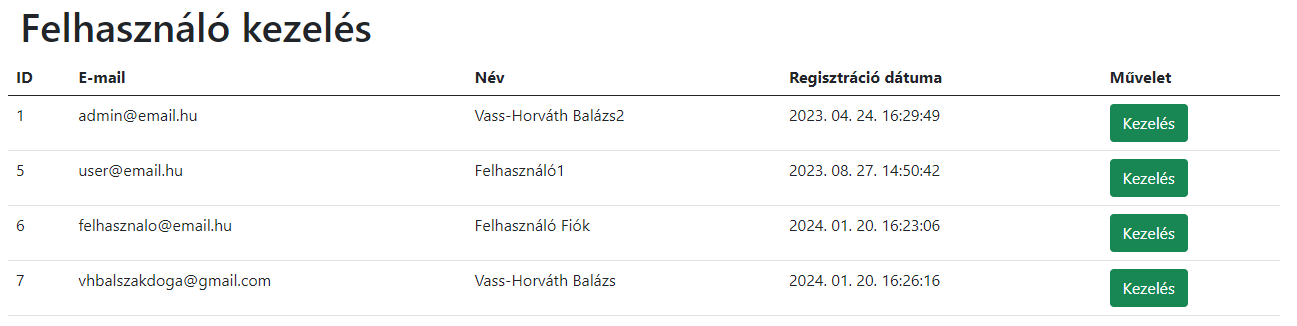
#### Felhasználó kezelés

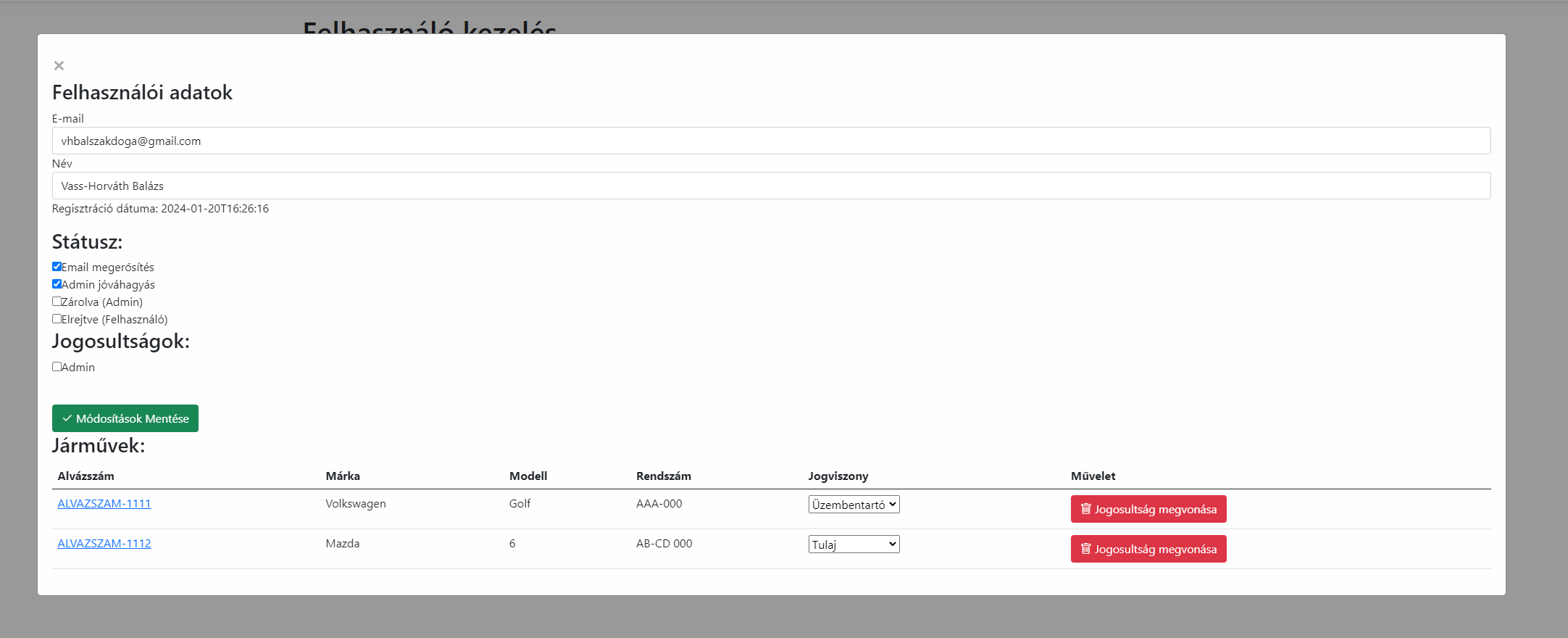
Adminként az egyik legfontosabb feladatunk lehet a felhasználók támogatása. A „Felhasználó kezelés” pont alatt megtekinthetjük az összes regisztrált felhasználót a rendszerben. Az előnézetben látjuk a felhasználó azonosítóját, email címét, nevét, és a regisztráció dátumát, továbbá találunk egy „Kezelés” gombot, mely megjeleníti a részletező nézetet egy felugró ablakban.

A felugró ablakban tudjuk módosítani a felhasználó email címét, nevét, és státuszait. Ezen túl itt adhatunk, vagy vehetünk el „Admin” jogosultságot is. Miután elvégeztük a kívánt módosításokat minden esetben kattintsunk a „Módosítások Mentése” gombra. Fontos, hogy minden esetben óvatosan, és körültekintően járjuk el az email cím módosításakor, ugyanis a rendszer csak az alkalmazás működését veszélyeztető hibákat szűri ki, egyéb logikai hibákat nem keres.

A felugró menü alján megjelennek a felhasználóhoz rendelt járművek, valamint az, hogy milyen jogosultsági szint köti össze őket. A táblázatos megjelenésben láthatjuk a járművek alvázszámát, márkáját, modelljét, rendszámát, továbbá az őket összekötő jogviszonyt. Amennyiben a jármű és a felhasználó között szeretnénk a jogviszonyt módosítani, egyszerűen csak válasszuk ki a lenyíló listából a kívánt jogviszonyt, vagy megvonáshoz kattintsunk a műveletek oszlopban lévő piros hátterű „Jogosultság megvonása” gombra. Jogviszony módosításakor nem szükséges a módosítások mentése gomb használata, ezek automatikusan érvénybe lépnek. A jármű alvázszámára kattintva új lapon megnyílik a járműkezelés admin felülete az adott járműre vonatkozóan.

Miután minden módosítást elvégeztünk, és szükség esetén mentettünk a felhasználóra vonatkozóan, kattintsunk a bal felső sarokban lévő „X”-re a felugró menü bezárásához. Amennyiben a mentést igénylő műveleteket nem mentjük, és úgy zárjuk be az ablakot, a módosítások elvesznek.

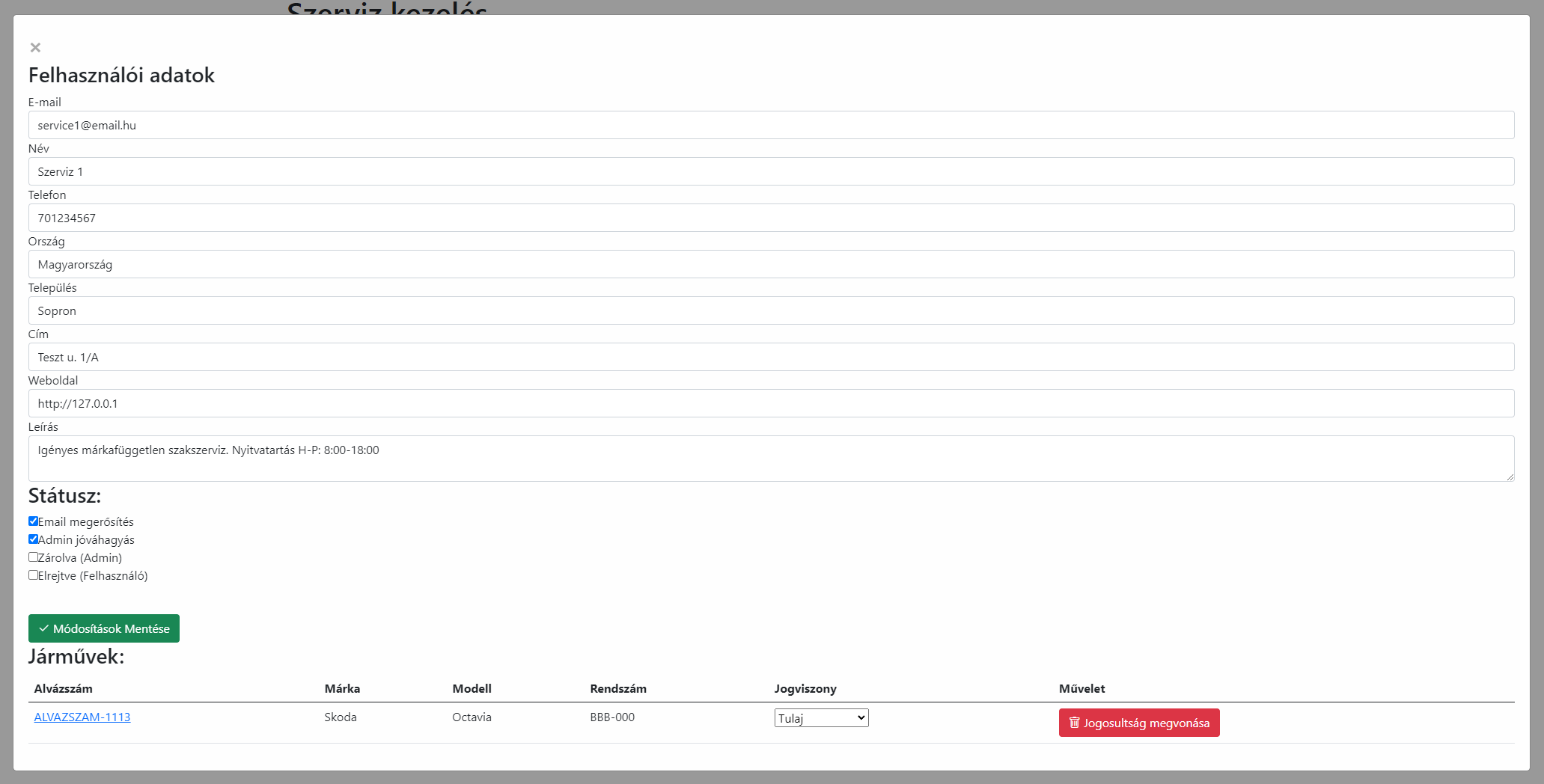




#### Szolgáltató kezelés

A szolgáltató kezelés nagyon hasonlóan működik, mint a hagyományos felhasználókezelés, valamint a funkciók megegyeznek szerviz, és kereskedő esetén is. Az admin menüben válasszuk ki a kívánt szolgáltató típusra vonatkozó kezelés opciót. Ekkor egy a felhasználókezelőhöz hasonló táblázat fog megjelenni, mely tartalmazza a szolgáltató azonosítóját, nevét, telephelyét (Város, település szinten), valamint telefonszámát. A szolgáltató adatainak módosításához itt is a „Kezelés gombra” kell kattintanunk, amely megjeleníti a részletező menüt. Itt módosíthatjuk a szolgáltató email címét, nevét, elérhetőségi adatait, weboldalát, valamint leírását. Szintén kezelhetjük a fiókra vonatkozó státuszokat, jogosultság rész azonban nem jelenik meg, ugyanis az csak hagyományos felhasználóhoz köthető. A menü alján a felhasználókezelőhöz hasonlóan itt is megjelennek a szolgáltató járművei, és a felhasználói menühöz hasonlóan módosíthatjuk a jogviszonyt, vagy új oldalon megnyithatjuk a járműre vonatkozó admin kezelő felületet.

A szervizek között továbbá szerepel egy kitűntetett szerepet betöltő rekord, mely 0-s azonosító alatt szerepel a rendszerben, és nem tartozik hozzá semmilyen más adat. Ennek célja, hogy adminisztrátorként rögzíthetünk szervizeseményt ez alatt a járműhöz akkor is, ha a karbantartást elvégző szerviz nem szerepel a rendszerben.



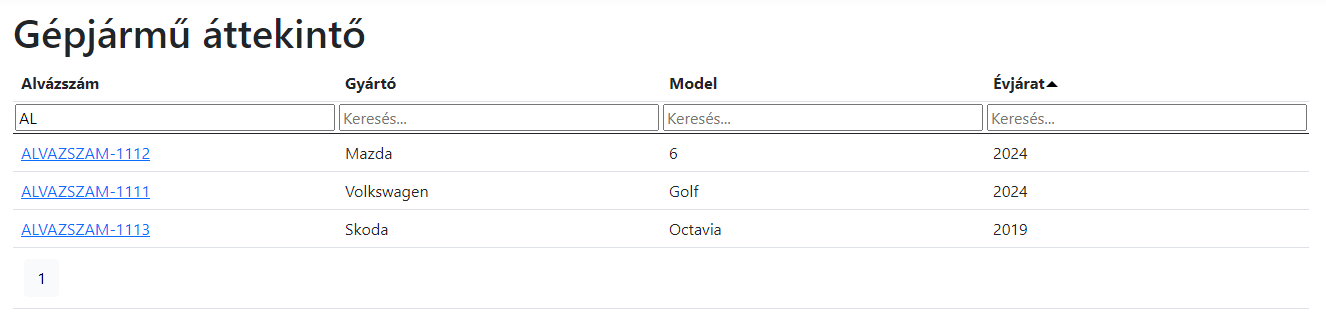
#### Jármű kezelés

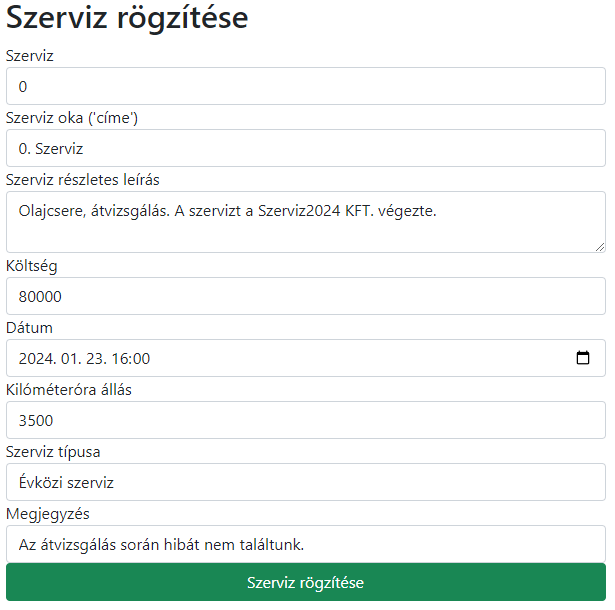
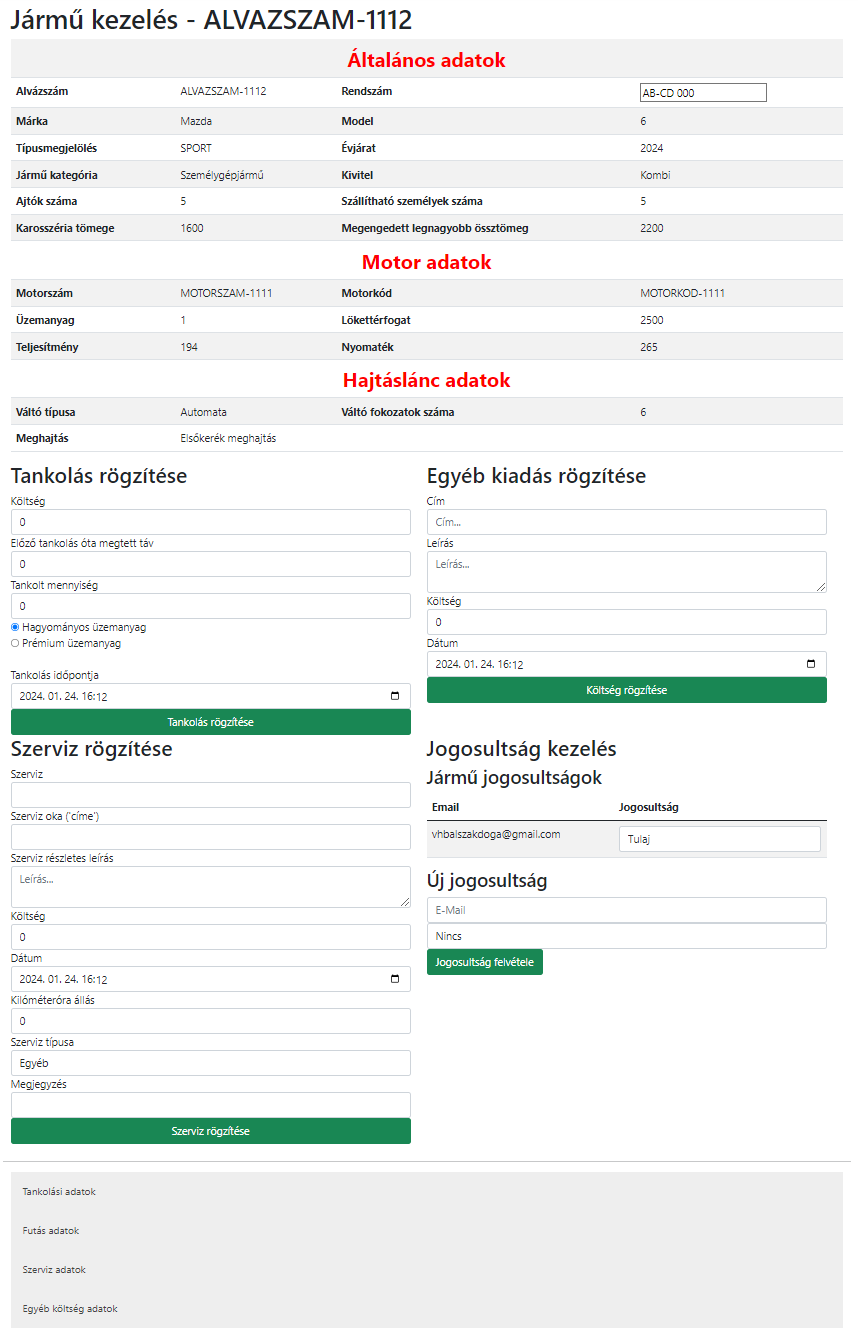
Az alkalmazás legfőbb célja, hogy a járművek életciklusát kövesse le, ezért elengedhetetlen, hogy igény esetén adminisztrátorként olyan módosításokat is elvégezhessünk, amit az átlag felhasználó nem tud. Az „Admin” menüpont alatti „Jármű kezelés” opciót választva az oldal elnavigál mindeket egy járműlistázó oldalra. Itt a rendszerbe rögzített összes jármű listázásra kerül. Lehetőségünk van a listában keresni alvázszám, gyártó, modell, és évjárat szerint. Továbbá a táblázatos nézetben egyidejűleg csak korlátozott számú jármű jeleníthető meg, így szükség esetén a táblázat alján megjelenő oldalszámokra kattintva tudunk lapozni. A táblázat fejlécére kattintva rendezhetjük a találatokat a kiválasztott oszlop alapján növekvő, vagy csökkenő sorrendbe. Az aktuális rendezési módot a fejléc mellett megjelenő nyíl jelzi.

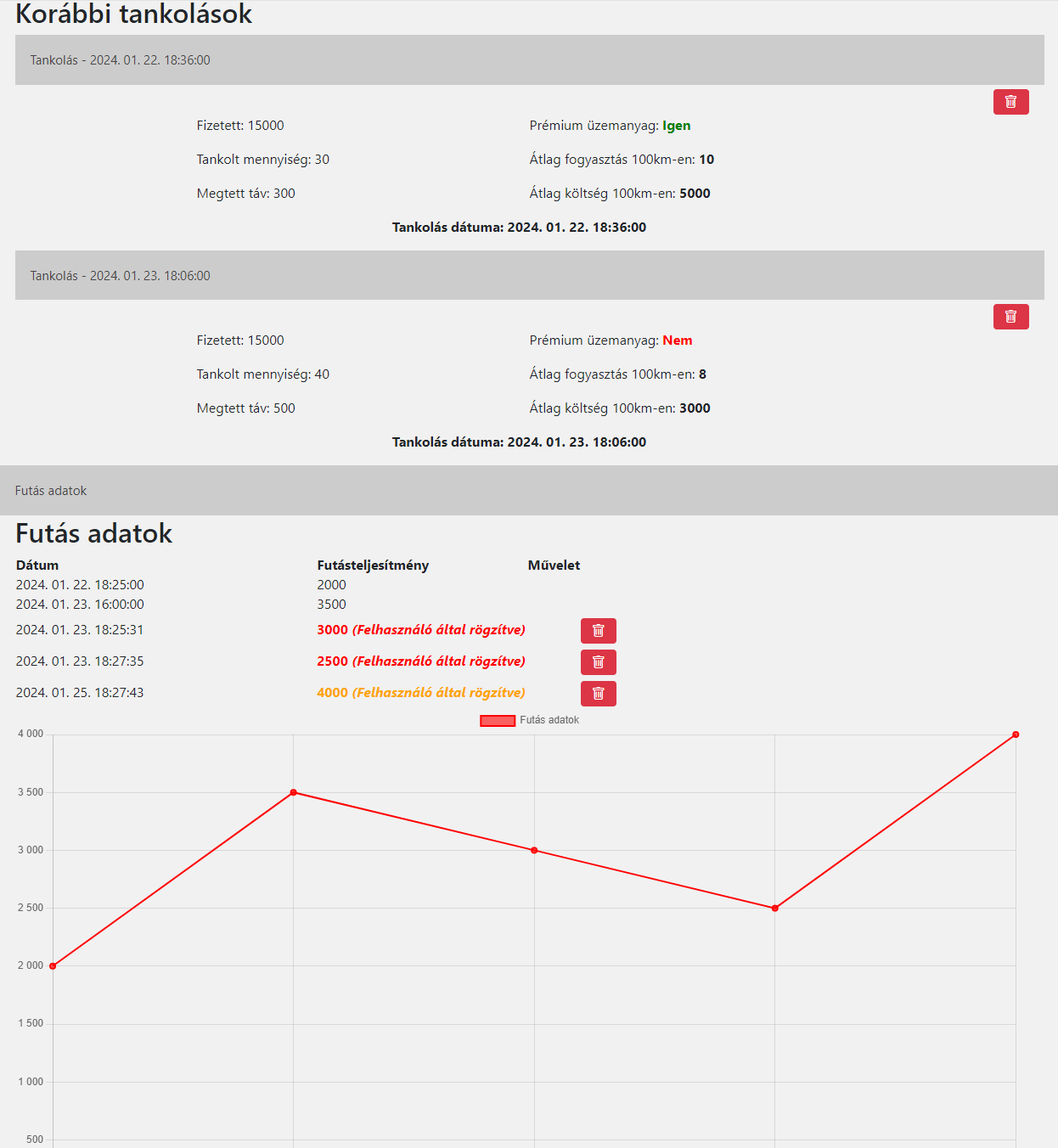
Egy jármű alvázszámára kattintva (kéken kiemelve) új lapon megnyithatjuk a járműre vonatkozó részletes admin felületet. Itt megjelenik a jármű minden adata a felhasználói nézethez hasonlóan. Mindazonáltal adminként lehetőségünk nyílik további funkciókra is, melyekre a tulajdonos felhasználónak sincs jogosultsága. Ide tartozik például a szerviz rögzítés, melyre akkor lehet szükség, ha a felhasználó olyan szervizben végeztette el a munkát, amely nincs regisztrálva a rendszerbe, vagy nem használja azt. Ezt a tulajdonosnak nem engedjük meg, ugyanis ezzel hamis képet festhetne a jármű karbantartásáról, adminisztrátorként viszont számla, vagy egyéb bizonyíték alapján eldönthetjük, hogy a felhasználó által mondott munkálatokat valóban elvégezték-e. Szervizesemény rögzítéshez meg kell adnunk a munkálatokat elvégző szervizt, vagy amennyiben nem szerepel a rendszerben, abban az esetben válasszuk a kitűntetett szerepet betöltő „0” azonosítójú „szervizt”, ebben az esetben a leírásban érdemes lehet jelölni, hogy valójában hol is végezték el azt. Adjuk meg a szerviz okát, mint például éves szerviz, töltsük ki a részletes leírásra vonatkozó mezőt. Adjuk meg a munkálatok költségét, illetve az elvégzés dátumát. Adminisztrátorként bármilyen korábbi, vagy későbbi dátumot megadhatunk, nem tiltja le a rendszer, így figyelmesen járjunk el a dátum megadásakor. További mezőink vannak a szerviz elvégzésekor lévő kilométer óraállás megadására, a szerviz típusának listából történő kiválasztására (például éves, évközi, műszaki vizsga, egyéb…), valamint megjegyzés írására. Miután minden adatot figyelmesen kitöltöttünk, és ellenőriztük azokat, kattintsunk a „Szerviz rögzítése” gombra.

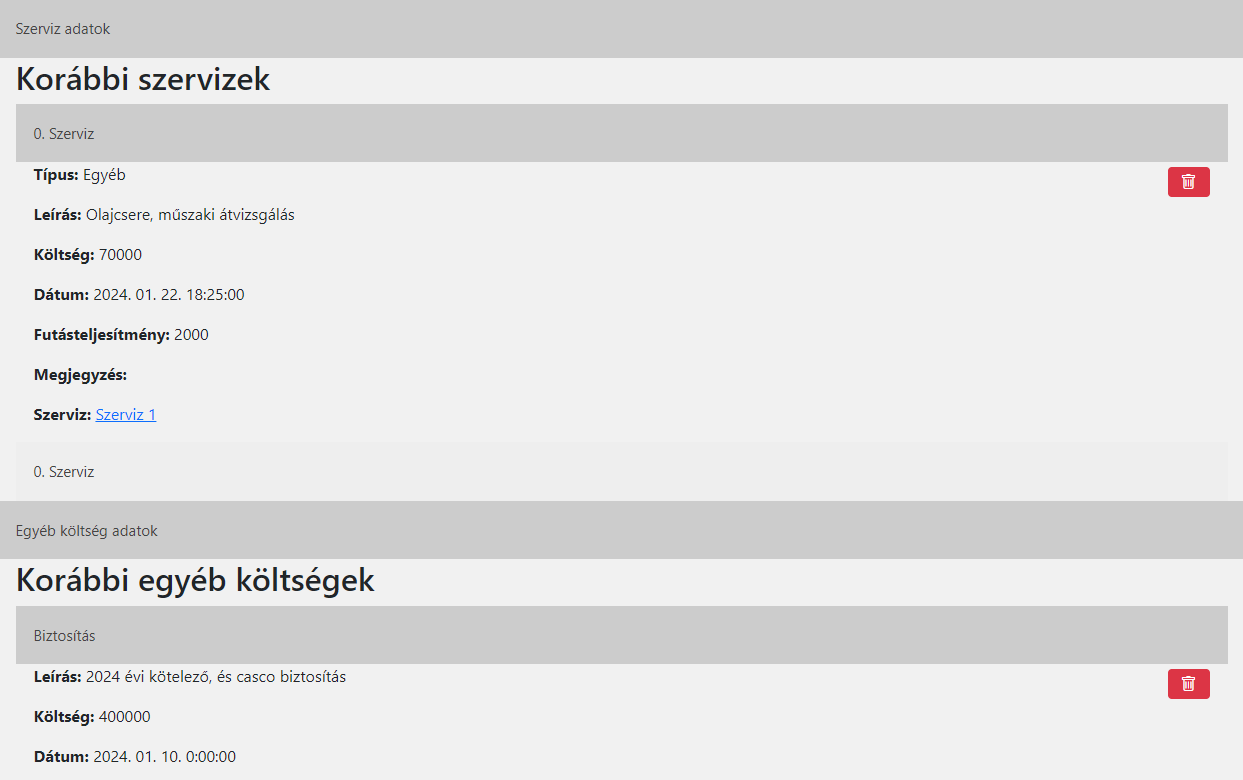
A tankolás, és egyéb kiadás rögzítés, továbbá a jogosultságkezelés a felhasználói felülethez hasonlóan működik.

Adminként továbbá lehetőségünk van a járműhöz kapcsolódó adatok törlésére beleértve a tankolásokat, futás adatokat, szervizeseményeket, valamint egyéb költségeket. Ezt az adat mellett található, piros hátterű, kuka piktogramot tartalmazó gomb megnyomásával tehetjük meg. A törlés funkciók szintén nem elérhetőek az átlagos felhasználónak, hogy ezzel ne tudják torzítani a jármű előéletét. Adminisztrátorként igazoltan indokolt esetben (például adminisztrációs hiba) azonban ezt megengedjük.









### Szerviz specifikus funkciók

### Kereskedő specifikus funkciók

### Gyártó specifikus funkciók

Gépjárműgyártók számára csak, és kizárólag API-n keresztüli kommunikációt biztosítunk. Az alkalmazás használatához az oldal adminisztrátoraival szükséges felvenni a kapcsolatot, akik API hozzáféréshez jogosító fiókot regisztrálnak a gyártónak.

A gyártók számára nyújtott szolgáltatások API dokumentációja a fejlesztői dokumentáció alatt található.