Informatikai rendszerterv

Molnár Ferenc

Mezőtúr, 2023

Tartalom

[1. Informatikai rendszerterv 4](#_Toc133477343)

[1.1 Bevezető, a probléma rövid ismertetése 4](#_Toc133477344)

[1.2 Választott téma indoklása 4](#_Toc133477345)

[1.3 Téma kifejtése 4](#_Toc133477346)

[2. Fogalmak és rövidítések 4](#_Toc133477347)

[3. Architektúra 5](#_Toc133477348)

[3.1 Áttekintés 5](#_Toc133477349)

[3.2 Rétegek és felelősségek 6](#_Toc133477350)

[3.3 Választott technológiák 17](#_Toc133477351)

[3.4 Asztali alkalmazás: 18](#_Toc133477352)

[3.5 Web-es app: 18](#_Toc133477353)

[3.6 Külső könyvtárak: 18](#_Toc133477354)

[3.7 Függőségek 20](#_Toc133477355)

[3.7.1 Minimális hardver követelmény: 20](#_Toc133477356)

[3.7.2 Ajánlott hardver követelmény: 20](#_Toc133477357)

[3.7.3 Desktop app: 20](#_Toc133477358)

[3.7.4 Web App: 20](#_Toc133477359)

[4. Modulok 21](#_Toc133477360)

[4.1 Áttekintés 21](#_Toc133477361)

[5. Perzisztencia 24](#_Toc133477362)

[5.1 A tárolás választott módja 24](#_Toc133477363)

[5.2 Koncepcionális adatmodell 24](#_Toc133477364)

[6. Folyamatok 27](#_Toc133477365)

[7. Felhasználói dokumentáció 28](#_Toc133477366)

[7.1 Általános látogató: 28](#_Toc133477367)

[7.1.1 Kezdőoldal: 28](#_Toc133477368)

[7.1.2 Navigációs sáv: 28](#_Toc133477369)

[7.1.3 Lábléc: 29](#_Toc133477370)

[7.1.4 Összes: 30](#_Toc133477371)

[7.1.5 Autó adatlapja: 30](#_Toc133477372)

[7.1.6 Keresés: 31](#_Toc133477373)

[7.1.7 Bejelentkezés: 33](#_Toc133477374)

[7.1.8 Elfelejtett jelszó: 34](#_Toc133477375)

[7.1.9 Regisztráció: 34](#_Toc133477376)

[7.1.10 Kedvencek: 35](#_Toc133477377)

[7.1.11 Adataim: 35](#_Toc133477378)

[7.1.12 Adatok módosítása: 36](#_Toc133477379)

[7.1.13 Kijelentkezés: 36](#_Toc133477380)

[7.2 Adminisztrátor: 36](#_Toc133477381)

[7.2.1 Felhasználói fiókok: 36](#_Toc133477382)

# Informatikai rendszerterv

Verzió: 1.0

## Bevezető, a probléma rövid ismertetése

Az autókereskedés eddig csupán facebookon működött, mert alig 100 autó volt eladó és hirdetett. 2022-ben az eddigi profit felhasználásával bővült a kínálat, több mint 300 autóra, amire már nem volt alkalmas a Facebook Marketplace. Ez okból kellett egy weboldal, és egy hozzátartozó adatbázis, mellyel könnyedén vezetni lehet az eladó autókat, és minden hozzájuk tartozó tulajdonságot.

## Választott téma indoklása

Gyerekkorom óta közel állnak hozzám az autók. Apukámmal mindig autót szereltünk, és így sok kellemes emlék köt össze a kocsikkal. Érdekel minden tulajdonságuk, felszerelésük, teljesítményük stb. Gyakran szoktam eladó autókat nézegetni, és kíváncsi voltam, hogy én magam milyen weboldalt tudnék erre létrehozni, és hogy milyen tudás áll egy ilyen weboldal mögött

## Téma kifejtése

A témám egy autókereskedés weboldala, ahol számon van tartva egy cég minden egyes eladott autója. Az oldalon megtalálható a cég elérhetősége, és az eladó autókról minden féle információ (ár, megtett kilométer, hengerfejek száma, szín, kivitel, évjárat, gyártó, modell, meghajtás, váltó). Az autókra ezek a tulajdonságok alapján szűrni is lehet. Be lehet állítani kedvenc autókat, hogy később ne kelljen keresgélni, hanem egy kattintással elérhető. Jelen pillanatban több mint 400 autó található meg az adatbázisban. Az oldalra regisztrálni is lehet, mint átlag felhasználó. Viszont vannak felhasználók, akik több jogosultsággal rendelkeznek, mint egy vendég, például az adminisztrátor, aki a regisztrált felhasználókat is meg tudja nézni. A weboldalnak van egy asztali alkalmazása is, ahol az adatbázis kezelésére szolgáló menük találhatóak meg. Ide is bejelentkezés alapján lehet csak bejutni különböző felhasználói szintek szerint.

# Fogalmak és rövidítések

* HTML5: A HTML (Hypertext Markup Language) átdolgozott változata, célja, hogy a webes alkalmazásokhoz ne kelljenek pluginok.
* MySQL: Adatbázis kezelő szerver.
* CSS: Olyan programnyelv, mellyel HTML weboldalakat dizájnolhatunk.
* EJS: Egy sablon nyelv, mellyel HTML oldalt lehet generálni egyszerű JavaScripttel
* JavaScript: Weboldalak programozási nyelve.
* .NET framework: Microsoft által létrehozott szoftver keretrendszer.

# Architektúra

## Áttekintés

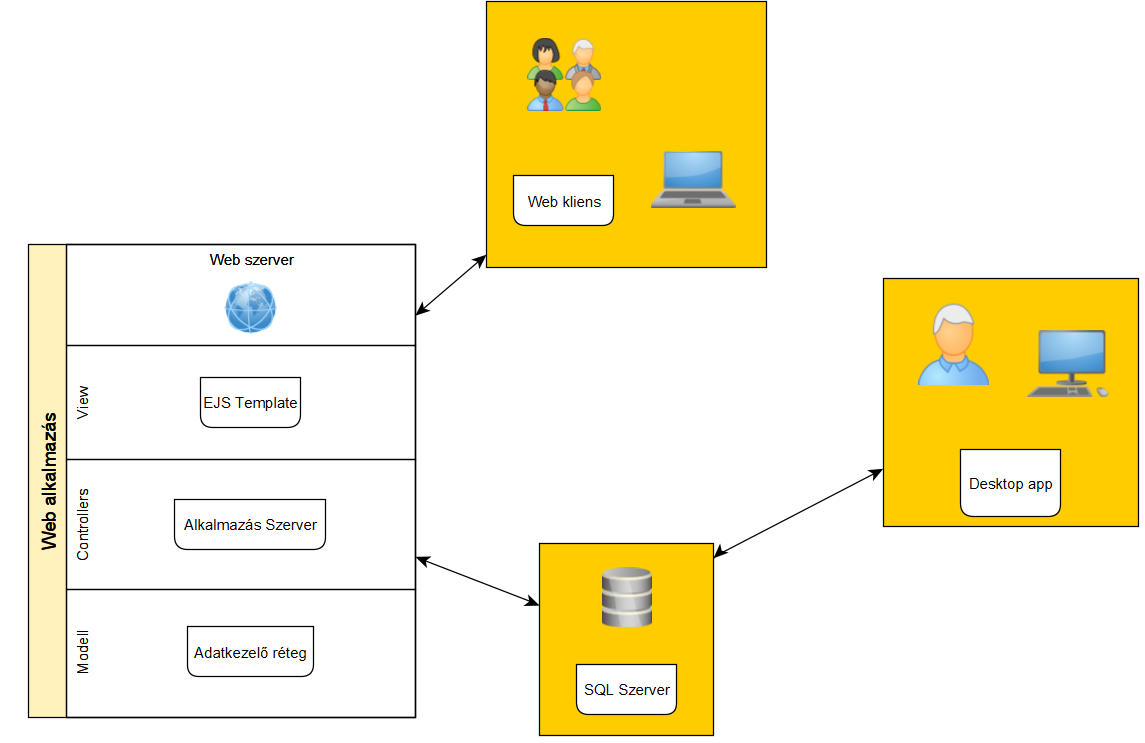
* **Asztali alkalmazás**
* **WEBkiszolgáló**
* **Müködési környezet**: Windows környezet
* Windows .NET app: Microsoft által készített keretrendszer, mely a gyors alkalmazás fejlesztést szolgálja.
* alkalmazásszerver – **NodeJS** v16.17.1: Webszerverek készítését szolgáló szoftverrendszer.
* **MySQL** adatbázisszerver 8.0.31: Adatbázis kezelésre szolgál
* **Connector/NET** 8.0.28
* Támogatott böngésző (Chrome, Firefox, Brave)
* Használat protokollok (http,https)
* **Portok**:
  + MySQL: 3306
  + http: 80 (Jelenleg a webkliens https protokollal működik)
  + https: **443**
* **SSL** tanúsítvány
  + Teszteléshez használt: OpenSSL programmal generált önaláírt tanúsítvány
  + Az átadott verzióhoz ajánljuk valamelyik tanúsítványkiszolgálótól megvásárolni a használt domainhez (pl.: carmarket.hu) tartozó tanúsítványt.
  + Tanúsítvány tároló mappa helye
  + [telepítési mappa]\cert\localhost\
  + [telepítési mappa]\bin\www.js tartalmazza a használt tanúsítvány elérhetőségét

const options ={

key:fs.readFileSync(path.join(\_\_dirname,'../cert/localhost/localhost.decrypted.key')),

cert:fs.readFileSync(path.join(\_\_dirname,'../cert/localhost/localhost.crt'))

}



1. ábra

## Rétegek és felelősségek

**Router Layer:**

A Router Layer tartalmazza az alkalmazás programozási felületének (API) útvonalait. Feladata, hogy meghívja az alkalmazás szerver megfelelő metódusát és várja a válaszát. Választ küldjön a szervertől a webes böngésző felé.

**A modul helye**: [telepítési mappa]\routes\ mappa

**A réteg komponensei:**

* *index.js* – általános útvonalak kezelése

|  |  |  |  |
| --- | --- | --- | --- |
| **URL** | **Paraméterek** | **Metódus** | **Funkció** |
| / | - | GET | Betölti a kezdőoldalt |
| /cars | - | GET | Betölti az összes autót |
| /cars/:page | page: oldalszám | GET | Egy sablont tölt be 30 autóval, a megadott oldalszámnak megfelelően |
| /search | - | GET | Betölti a keresési oldalt |
| /search | - | POST | A bodyból kiveszi a szükséges paramétereket/feltételeket, és aszerint visszaadja a keresésnek megfelelő elemeket |
| /car/:id | id: autó azonosítója | GET | Azonosító alapján betölti az autó adatlapját |

* *users.js* – a felhasználó kezeléshez tartozó útvonalak kezelése

|  |  |  |  |
| --- | --- | --- | --- |
| **URL** | **Paraméterek** | **Metódus** | **Funkció** |
| /user/login | - | GET | Betölti a bejelentkezési oldalt |
| /user/login | - | POST | A bodyból kiveszi a szükséges paramétereket/feltételeket, és aszerint bejelentkezteti a felhasználót |
| /user/registration | - | GET | Betölti a regisztrációs oldalt |
| /user/registration | - | POST | A bodyból lekért paraméterek alapján regisztrálja a felhasználót |
| /user/accounts | - | GET | Betölti a felhasználók listáját |
| /user/profile | - | GET | Betölti a felhasználó adatait |
| /user/favourites | - | GET | Betölti a felhasználó kedvenc autóit |
| /user/edit | - | GET | Betölti az oldalt, ahol a felhasználó szerkesztheti az adatait |
| /user/edit | - | POST | Betölti az oldalt, ahol a felhasználó szerkesztheti az adatait |
| /user/addFavourute | - | POST | A bodyból kiszedi az autó azonosítóját, és hozzáadja a felhasználó kedvenceihez |
| /user/removeFavourite | - | POST | A bodyból kiszedi az autó azonosítóját, és eltávolítja azt a felhasználó kedvenceiből |
| /user/reset-password | - | GET | Megjeleníti az új jelszó generálására szolgáló weblapot. |
| /user/reset-password | - | POST | Leellenőrzi, hogy a megadott email cím megtalálható-e az adatbázisban, ha nem akkor megjeleníti az oldalt újra, egy hibaüzenettel, ha igen, akkor generál egy 10 karakter hosszú véletlenszerű jelszót, és azt a emailjs fetch API segítségével, kiküldi a felhasználó email címére. |
| /user/logout | - | GET | Kijelentkezteti a felhasználót |

**Controller réteg:**

A Controller Layer kezeli az alkalmazás üzleti logikáját. Ez azt jelenti, hogy az adatokat átalakítják vagy kiszámítják, hogy megfeleljenek az adatbázismodell követelményeiknek, mielőtt elküldök őket a kiszolgálóra. A router réteg része.

**Adatkezelő réteg:**

Az adatkezelő réteg (Modells) hozzáféréssel rendelkezik az adatbázishoz, hogy adatokat hozzon létre, töröljön vagy szerkesszen.

Az adatbázishoz szükségek paramétereket a **[telepítési mappa]\db\dbconfig.js** fájl tartalmazza:

const config = {

    connectionLimit: 10,

    password: 'root',

    user: 'root',

    database: 'auto',

    host: 'localhost',

    port: 3306,

    dateStrings: true

  };

A telepítést követően a paramétereket a helyi sajátosságoknak megfelelően kell beállítani.

A telepített verziónál ajánlatos egy korlátozott jogosultságokkal rendelkező felhasználó létrehozása a MySQL szerveren és ezzel a felhasználóval csatlakozni.

**connectionLimit:** Megadja mennyien csatlakozhatnak egyszerre a webkliensen keresztül az adatbázishoz.

**password**: a felhasználóhoz tartozó jelszó.

**user**: a MySQL szerverhez kapcsolódó felhasználó neve.

**database:** Az adatbázis neve.

**host**: amennyiben az SQL szerver és a webkiszolgáló külön gépen kerül telepítésre, itt kell megadni a MySQL szervergép hostnevét

**port:** A port amin a MySQL szerver fut.

**dateStrings**: Dátumok könnyebb kezelése.

**A réteg felelősségei:**

Ez az a réteg, ahol a szerver logikájából érkező összes kérést és választ kezelik. Az adatelérési (Data Access Layer) metódusok változókat kapnak a controller rétegtől. Ezek a változók szükségesek az SQL lekérdezésekhez. Az SQL szervertől kapott válasz alapján ígéretet (Promise) küld, amely egy objektum. Az objektum a szerver válaszának rendelkezésre állása után kerül meghatározásra. Ha a kérés sikeres, a DAL ígéretet egy objektum oldja fel, amely visszatér a szolgáltatási réteghez, amely maga is visszatér az útválasztási réteghez. Amikor az útválasztási réteg megkapja a szolgáltatási réteg által visszaadott objektumot, az útválasztási réteg JSON formátumban küldi el az objektumot a View rétegnek.

**A modul helye: [telepítési mappa]\db\dboperation.js**

**A réteg komponensei:**

|  |  |  |
| --- | --- | --- |
| **Név** | **Paraméterek** | **Funkció** |
| Select30 | -pageNo: oldalszám | Lekérdezi 30 autó összes adatát, annyival eltolva amennyi az oldalszám |
| SelectAll | - | Lekérdezi az összes autó adatát az adatbázisban |
| SelectOne | -carId | Lekérdezi egy autó összes adatát, az autó azonosítója alapján |
| SelectAllSpecial | - | Lekérdezi a különleges ajánlatokban lévő autók azonosítóját, gyártóját, modellét, évjáratát, és a hozzáírt bemutató szöveget |
| CountElements | - | Lekérdezi, hogy hány autó adata szerepel az adatbázisban |
| SelectAllMake | - | Lekérdezi az összes gyártót |
| SelectAllCondition | - | Lekérdezi az összes állapotot |
| SelectAllColor | - | Lekérdezi az összes színt |
| SelectAllFuel | - | Lekérdezi az összes üzemanyag típust |
| SelectAllType | - | Lekérdezi az összes kivitelt |
| SelectAllTransmission | - | Lekérdezi az összes váltótípust |
| SelectAllDrive | - | Lekérdezi az összes meghajtást |
| SelectModelByMake | - | Lekérdezi az összes modellt és gyártót, a gyártó szerint csoportosítva |
| SelectMaxPrice | - | Lekérdezi a legdrágább autó árát |
| SelectMinYear | - | Lekérdezi a legrégebbi autó évjáratát |
| SelectMaxYear | - | Lekérdezi a legújabb autó évjáratát |
| EditUser | -name: felhasználó módosított neve  -email: felhasználó módosított email címe  -username: felhasználó módosított felhasználóneve  -id: felhasználó azonosítója | A megadott azonosítóval rendelkező felhasználóhoz tartozó mezőket módosítja a paramétereknek megfelelően |
| VerifyUser | -username: felhasználónév  -password: jelszó | Megnézi, hogy található-e az adatbázisban a paraméterekben megadott felhasználónév-jelszó páros. A jelszót titkosítani kell a lekérdezés során |
| VerifyUserById | -id: felhasználó azonosítója  -password: felhasználó által megadott jelszó | Megnézi, hogy található-e az adatbázisban a paraméterekben megadott azonosító-jelszó páros. A jelszót titkosítani kell a lekérdezés során. |
| CheckEmail | -email: felhasználó által megadott email cím | Lekérdezi az összes olyan rekordot, ahol az email cím megfelel a paraméterben megadottban. |
| NewPassword | -email: megadott email cím  -password: az újonnan generált jelszó. | Email cím alapján módosítja a jelszót. |
| NewUser | -username: felhasználónév  -password: jelszó  -name: teljes név  -email: email cím  -letters: hírlevelekre feliratkozik-e | Beilleszt az adatbázisba egy új felhasználót a megadott paramétereknek megfelelően. Jogosultsági szintje automatikusan 1 lesz. |
| AllUsers | - | Lekérdezi az összes felhasználó összes adatát |
| SelectUser | -userId: felhasználó azonosítója | Lekérdezi egy felhasználó összes adatát, a megadott azonosító szerint. |
| CheckUsed | -username: felhasználónév  -email: email cím | Lekérdezi, az összes olyan felhasználó adatát, akinél a megadott felhasználónév, vagy email cím szerepel. |
| NewFavorite | -userId: felhasználó azonosítója  -carId: autó azonosítója | A favorites táblába beilleszt egy új sort a paramétereknek megfelelőlen. |
| RemoveFavorite | -userId: felhasználó azonosítója  -carId: autó azonosítója | A favorites táblábl kitörli a paramétereknek megfelelő sort. |
| Favorites | -userId: felhasználó azonosítója | Lekérdezi a megadott felhasnzálóhoz tartozó összes kedvencnek jelölt autót |
| CheckFavorite | -userId: felhasználó azonosítója  -carId: autó azonosítója | Lekérdezi a paramétereknek megfelelő autókat a favorites táblából. |
| Filter | -make: gyártó  -models: tömb, modelleket tartalmazza  -years: tömb, az évjárat alsó és felső határát tartalmazza  -prices: tömb, az ár alsó és felső határát tartalmazza  -conditions: tömb, állapotokat tartalmaz  -colors: tömb, színeket tartalmaz  -fuels: tömb, üzemanyag típusokat tartalmaz  -types: tömb, kiviteleket tartaémaz  -transmissions: tömb, váltókat tartalmaz  -drives: tömb, meghajtásokat tartalmaz | A megadott paramétereknek megfelelően visszaadja az összes olyan autó összes adatát, amelyek megfelelnek a feltételeknek. |

**View réteg:**

A html oldal előállításáért (render) az EJS template engine a felelős.

**A modul helye:** [telepítési mappa]\views mappa

**Réteg felelőssége:**

A JSON formátumban megkapott paraméterek és az EJS fájlban meghatározott sablon alapján előállítja a kimeneti HTML formátumot. A generált HTML oldal kódját visszaadja az útválasztási rétegnek.

**A réteg komponensei:**

|  |  |  |
| --- | --- | --- |
| **Neve** | **Szükséges bemenet** | **Funkció** |
| 30scroll.ejs | -list: tömb, ami tartalmazza az autók adatait | Betölti annyi autó kártyáját, amennyit megkap |
| accounts.ejs | -list: tömb, mely tartalmazza a felhasználók adatait | Felhasználók listájának megtekintése |
| allList.ejs | - | Összes autó listájának a váza, majd a 30scroll.ejs és „/cars/:page” url segítségével, tölti be az autókat harmincasával, ahogyan görget le a felhasználó |
| car.ejs | -images: tömb, mely tartalmazza az autóhoz tartozó képek neveit  -list: tömb, melynek egy eleme van, mely az adott autóhoz tartozó adatokat tartalmazza  -favourite: egy változó mely azt tartalmazza, hogy az autó hozzá van-e adva a felhasználó kedvenceihez, alapesetben hamis | Autóhoz tartozó adatok megjelenítése |
| editProfile.ejs | -lastname: változó, tartalmazza a felhasználó vezetéknevét  -firstname: felhasználó keresztneve  -email: felhasználó e-mail címe  -username: felhasználó felhasználóneve | Itt tudja a felhasználó megtekinteni az adatait |
| emailSent.ejs | -email: az email cím, amire el lett küldve az email. | Itt a felhasználó kap egy megerősítést, hogy az email el lett küldve, melyben szerepel az új jelszó. |
| error.ejs | -msg: objektum, mely tartalmazza a hiba részleteit | Ha a szerver hibára fut, itt megjeleníti |
| favourites.ejs | -list: tömb, mely tartalmazza a felhasználóhoz tartozó kedvenc autókat | Megjeleníti a felhasználó kedvence autóit |
| footer.ejs | - | A láblécet tartalmazza |
| getEmail.ejs | -emailError: alapértelmezett érteke hamis, akkor lesz igaz, ha a felhasználó rossz email címet ad meg, amely nem szerepel az adatbázisban | Egy felület, ahol az új jelszó kéréséhez a felhasználó meg tudja adni a jelszavát. |
| icon.ejs | - | A weboldalhoz tartozó ikon megjelenítéséért felel |
| index.ejs | -specials: tömb, mely a különleges ajánlatok autót tartalmazza | A kezdőlapot jeleníti meg. |
| login.ejs | -loginError: alapesetben hamis, akkor lesz igaz, ha a felhasználó rossz adatokat adott meg bejelentkezésnél | Megjeleníti a bejelentkezési oldalt. |
| navbar.ejs | -allPage: ha az Összes oldalon van a felhasználó, akkor igaz, amúgy hamis  -searchPage: ha a Keresés oldalon van a felhasználó, akkor igaz, amúgy hamis  -loggedIn: Be van-e jelentkezve a felhasználó, ha igen igaz, amúgy hamis  -username: A bejelentkezett felhasználó felhasználóneve  -permission: A bejelentkezett felhasználó jogosultsági szintjét tartalmazza  -loginPage: ha a Bejelentkezés fülön van a felhasználó, akkor igaz, amúgy hamis | A navigációs sáv megjelenítésért felel. |
| profile.ejs | -list: tömb, mely egy felhasználó adatait tartalmazza | Megjeleníti a felhasználóhoz tartozó adatokat. |
| registration.ejs | -usedData: változó, alap esetben hamis, akkor lesz igaz, ha a felhasználónév vagy email cím már foglalt  -lastname: változó, a regisztráló felhasználó vezetéknevét tartalmazza  -firstname: változó, a regisztráló felhasználó keresztnevét tartalmazza  -password: változó, a regisztráló felhasználó jelszavát tartalmazza  -passwordAgain: változó, a regisztráló felhasználó ismételten beírt jelszavát tartalmazza | Megjeleníti a regiszrációhoz szükséges űrlapot. Hibás adatok esetén, a felhasználónak nem kell mégegyszer megadnia minden adatot. |
| search.ejs | -modelsWithMake: egy tömb, mely a kiválasztott márkához tartozó modelleket tartalmazza  -makes: összes gyártót tartalmazó tömb  -minYear: a legkisebb évjáratú autó évjárata  -maxYear: a legkésőbbi évjáratú autó évjárata  -maxPrice: a legdrágább autó ára  -conditions: az összes állapotot tartalmazó tömb  -colors: az összes színt tartalmazó tömb  -fuels: az összes üzemanyagot tartalmazó tömb  -types: az összes kivitelt tartalmazó tömb  -transmissions: az összes váltót tartalmazó tömb  -drives: az összes meghajtást tartalmazó tömb | Az autók szűréséhez szolgáló űrlapot tölti be. |

## Választott technológiák

Miért ezt választottam?

* Web-es technológia: Több felhasználó tudja egyszerre használni (akár több száz). Aszinkron programozási technikát is azért választottuk, mert nagy számú felhasználó esetén nem várja meg az adatbázis válaszát, hanem egyidejűleg futtatja a lekérdezéseket. Másik előnye, hogy az interneten keresztül bárhonnan el lehet érni a szolgáltatást.
* Desktop: Korlátozott számú felhasználó használja csak az adminisztrátori funkciókat (maximum 2-3), de ez is több felhasználós, így egyszerre többen is hozzá tudnak férni az adatbázishoz. Viszont elegendő az, ha a lokális hálózaton elérhető.

## Asztali alkalmazás:

* Fejlesztői környezet:
  + Visual Studio
  + MySQL for Visual Studio
* Programnyelv: C#
* Operációs rendszer: Windows
* Keretrendszer: .Net framework 4.8 Advanced Services
* ADO.NET programkönyvtár
* Connector/NET for MySQL 8.0.28 x86

## Web-es app:

* Fejlesztői környezet:
  + NodeJS
* Programnyelv: Javascript
* Operációs rendszer: Windows/Linux

## Külső könyvtárak:

* **express:** *~4.16.1*, webkiszolgáló keretrendszere
  + Az express egy webalkalmazás keretrendszer, amely széles körű funkciókat kínál webes és mobilalkalmazások létrehozásához. Segít a szerverek és útvonalak kezelésében.
* **express-session:** *^1.17.3*, session kezelés
  + egy HTTP szerveroldali keretrendszer, amelyet munkamenet köztes szoftver létrehozására és kezelésére használnak.
* **bootstrap:** *^5.2.3*, bootstrap kezelés
  + A bootsrap egy népszerű CSS-keretrendszer
* **cookie-parser**: *~1.4.4*,
  + A cookie-parser egy köztes szoftver, amely elemzi az ügyfél kérési objektumához csatolt cookie-kat.
* **cors:** ^2.8.5, Cross-Origin Resource Sharing
  + Segítségével a front-end tud kéréseket küldeni egy külső back-end szervernek. Az emailjs-hez kell.
* **debug:** *~2.6.9*, fejlesztéshez
  + Olyan folyamat amely azonosítja a hibákat.
* **ejs:** *^3.1.8*,
  + Ez egy egyszerű sablonnyelv/motor, amely lehetővé teszi a felhasználó számára, hogy egyszerű javascripttel HTML-t generáljon.
* **https:** *^1.0.0*,
  + https protokoll használatatát teszi lehetővé
* **http-errors:** *~1.6.3*,
  + Http hibák eredetének kinyomozásában segít.
* **morgan:** *~1.9.1*,
  + A Morgan egy HTTP kérés szintű köztes szoftver. Ez egy nagyszerű eszköz, amely naplózza a kéréseket néhány egyéb információval együtt a konfigurációjától és a használt előre beállított értéktől függően. Nagyon hasznosnak bizonyul a hibakeresés során, és akkor is, ha naplófájlokat szeretne létrehozni.
* **mysql2:** *^3.0.1*,
  + A MySQL adatbázis eléréséhez szolgáló illesztő könyvtár.
* **node-fetch:** ^2.6.9
  + fetch API kérés indítása JavaScriptből.
* **nodemon:** *^2.0.20*,
  + Amikor az éppen futó kódunkon végzünk el változtatásokat, a mentést követően nem lépnek érvénybe ezen változtatások. Minden egyes alkalommal újra kell indítani a teljes programot. Komplexebb applikációk esetén már egy csomó időt elvesz a fejlesztőktől. A probléma megoldására alkották meg a Nodemon modult, mely minden módosítás után automatikusan újraindítja a szervert.
* **path:** *^0.12.7*,
  + olyan karakterlánc, amely egy címtárszerkezetben lévő hely egyedi azonosítására szolgál.
* **fs:** *^0.0.1-security*,
  + mappákból való fájl beolvasást teszi lehetővé

## Függőségek

MySQL Server 8.0.27 adatbázis szerver

### Minimális hardver követelmény:

* Processzor: 1 gigahertzes (GHz) vagy gyorsabb processzor
* RAM: 1 gigabájt (GB) 32 bites rendszerhez vagy 2 GB 64 bites rendszerhez
* Merevlemez-terület: 16 GB 32 bites operációs rendszerhez vagy 20 GB 64 bites operációs rendszerhez
* Videokártya: A DirectX 9-es vagy újabb verziójával kompatibilis, WDDM 1.0 illesztőprogrammal
* Kijelző: 1024x600 vagy nagyobb felbontás

### Ajánlott hardver követelmény:

* Processzor: 2.5 gigahertzes (GHz) vagy gyorsabb processzor
* RAM: 2 gigabájt (GB) 32 bites rendszerhez vagy 4 GB 64 bites rendszerhez
* Merevlemez-terület: 20 GB 32 bites operációs rendszerhez vagy 30 GB 64 bites operációs rendszerhez
* Videokártya: A DirectX 9-es vagy újabb verziójával kompatibilis, WDDM 1.0 illesztőprogrammal
* Kijelző: 1920x1080 vagy nagyobb felbontás

### Desktop app:

* Operációs Rendszer: Windows 10
* Keretrendszer: .NET framework 4.8 Advanced Services

### Web App:

* Windows/Linux
* NodeJS
* Teljes mappa struktúra
* package.json állományban megtalálható függőségek telepítése (npm install vagy a mellékelt node\_modules mappa)
* tanúsítvány, amennyiben van saját

# Modulok

## Áttekintés

A szoftver architektúrája az alábbi modulokra, komponensekre oszlik.

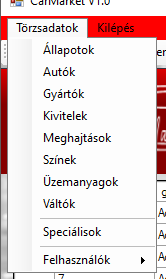
A rendszer két fő modulból áll:

* Desktop alkalmazás: Adminisztrátor funkciók megvalósítására szolgáló alkalmazás. A rendszerben használt törzsadatok kezelésére szükséges felületet szolgáltatja.
  + Törzsadatok modul:
    - Állapotok:
    - Autók
    - Gyártók
    - Kivitelek
    - Meghajtások
    - Színek
    - Üzemanyagok
    - Váltók
    - Speciálisok
  + Felhasználók modul
    - Felhasználók
    - Kedvencek
  + Alkalmazás működése:

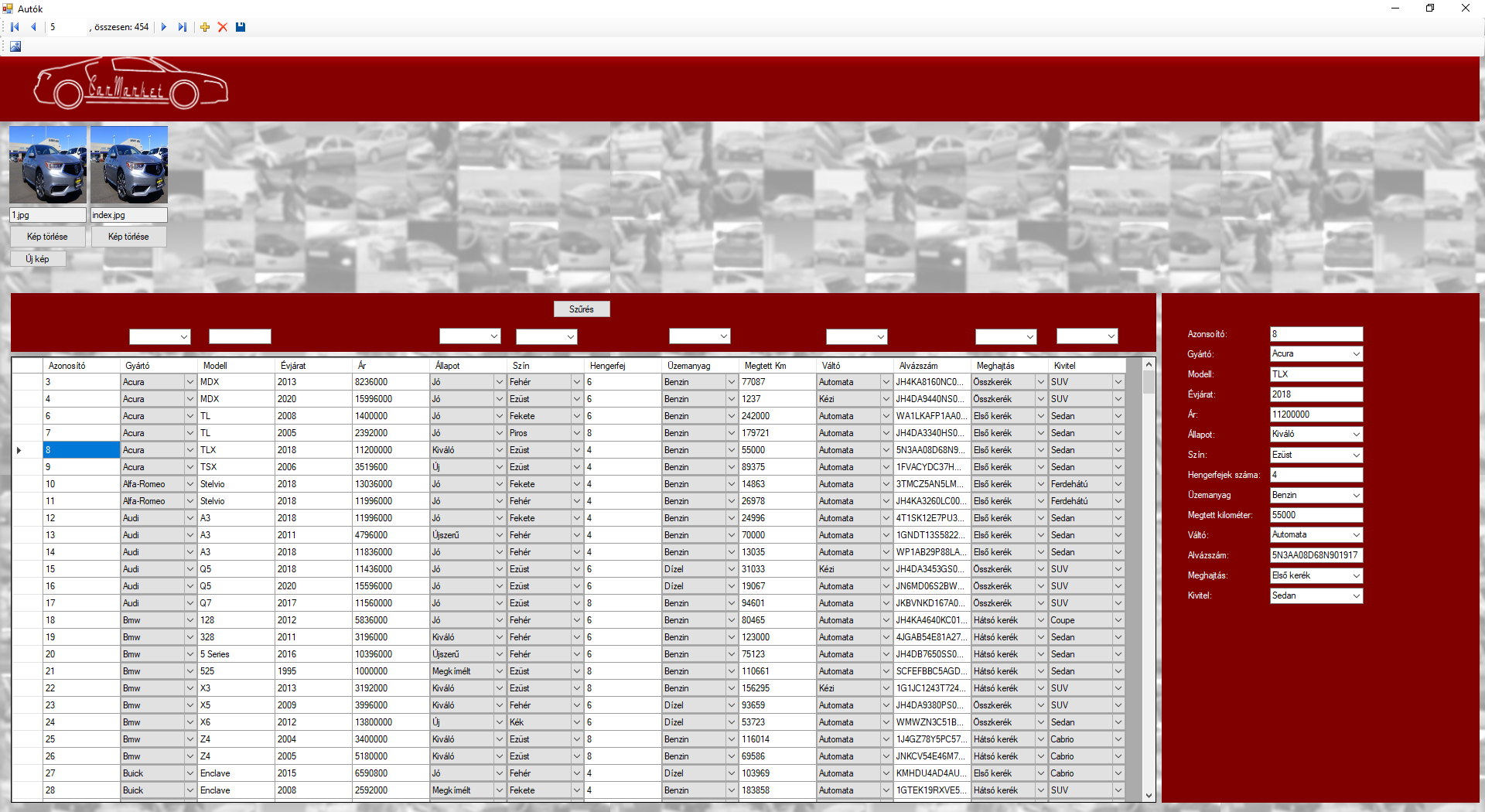
Az alkalmazás elindításakor egy bejelentkezési ablak ugrik be. Itt meg kell adni a felhasználónevet és a jelszót. Sikeres bejelentkezés esetén az ablak bezárul.



Az alkalmazásban az adatbázisban található összes táblát meg lehet tekinteni a menüsorban való navigálás segítségével. A Törzsadatok fül lenyitásánál megjelenik az összes tábla.



A sok tábla közül, az Autók táblában tud az adminisztrátor módosításokat végezni.



Ha a listában egy új autót jelöl ki, annak fent megjeleníti az összes képét. A képeket lehet törölni a Kép törlése gombbal, de lehet felvinni új képeket is az Új kép gombbal. Egyszerre akár több képet is kiválaszthat és tölthet fel. A kijelölt autó adatait a jobb oldali menüben tudja módosítani.

Szűrni is lehetséges a lista fölötti kis menüben. Miután beállította a megfelelő értékeket a Szűrés gomb megnyomása után a listában már csak a feltételeknek megfelelő autók fognak látszódni.

A menüsornál a kék gombra kattintva ki tudja nyomtatni a listában található autókat.

* WEBkiszolgáló:
  + Termékmegjelenítő: Megjeleníti az összes eladó autót.
  + Kedvencek:
    - A regisztrált felhasználó által kiválasztott autók rögzítésre kerülnek az adatbázisban, mint az ő kedvencei.
  + Szűrés:
    - Különböző feltételek alapján lehet szűrni az autók között.
  + Felhasználó kezelés:
    - Bejelentkezés: Felhasználónév-jelszó párossal lehet bejelentkezni.
    - Regisztráció: A szükséges adatok megadása után a felhasználó adatai el lesznek mentve az adatbázisban.
    - Adatok módosítása: A kért adatok megadása után a felhasználó tudja módosítani az adatait.
    - Új jelszó generálása: Elfelejtett jelszó esetén a felhasználó email címére kiküld egy új generált jelszót.
  + Különleges ajánlatok:
    - Adatbázisban a specials nevű táblában tárolódik az autó azonosítója, és hozzá a leírás. Amely autók megtalálhatók a táblában, azok jelennek meg a főoldalon.

# Perzisztencia

Adattárolás:

* MySQL adatbázis
* Képek tárolása a fájlrendszer mappáiban

## A tárolás választott módja

**MySQL:**

A MySQL egy többfelhasználós, többszálú, SQL-alapú relációs adatbázis-kezelő szerver. Többféle platformon futtatható. Képes az adatok táblázatokban történő tárolására, kezelésére és megjelenítésére. Kliens-szerver rendszerként működik.

**Fájlok/képek tárolása:**

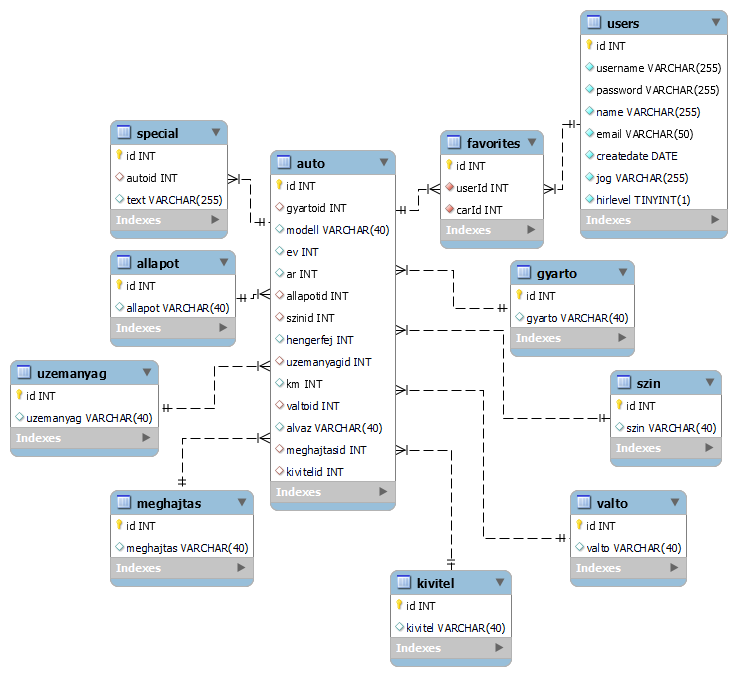
A képek autónként külön mappába tárolódnak.

Az autók képeit a webes alkalmazás telepített mappájában tároljuk, helye: [telepítési mappa]\public\src\database\[autó azonosítója]\

Minden autóhoz tartozik egy index kép, és további nagyobb felbontású képek.

## Koncepcionális adatmodell

ERD diagram:



Táblák:

* **auto**: Ez a főtábla.
  + id: Az autó egyedi azonosítója. Egész szám. Elsődleges kulcs.
  + gyartoid: Az autó gyártójának az azonosítója. Egész szám. Idegen kulcs.
  + modell: Az autó modellje. Szöveg.
  + ev: Az autó évjárata. Egész szám.
  + ar: Az autó ára forintban. Egész szám.
  + allapotid: Az autó állapotának az azonosítója. Egész szám. Idegen kulcs.
  + szinid: Az autó színének az azonosítója. Egész szám. Idegen kulcs.
  + hengerfej: Az autó hengerfejeinek a száma. Egész szám.
  + uzemanyagid: Az autó működéséhez szükséges üzemanyag azonosítója. Egész szám. Idegen kulcs.
  + km: Az autó kilométeróra állása. Egész szám.
  + valtoid: Az autó váltó típusának az azonosítója. Egész szám. Idegen kulcs.
  + alvaz: Az autó alvázszáma. Szöveg.
  + meghajtasid: Az autó meghajtásának azonosítója. Egész szám. Idegen kulcs.
  + kivitelid: Az autó kivitelének az azonosítója. Egész szám. Idegen kulcs.
* **gyarto:**
  + id: A gyártó/márka egyedi azonosítója. Egész szám. Elsődleges kulcs.
  + gyarto: A gyártó/márka neve. Szöveg.
* **allapot:**
  + id: Az állapot egyedi azonosítója. Egész szám. Elsődleges kulcs.
  + allapot: Az állapot megnevezése. Szöveg.
* **szin:**
  + id: A szín egyedi azonosítója. Egész szám. Elsődleges kulcs.
  + szin: A szín megnevezése. Szöveg.
* **uzemanyag:**
  + id: Az üzemanyag típus egyedi azonosítója. Egész szám. Elsődleges kulcs.
  + uzemanyag: Az üzemanyag fajta megnevezése. Szöveg.
* **valto:**
  + id: A váltó típus egyedi azonosítója. Egész szám. Elsődleges kulcs.
  + valto: A váltó típus megnevezése. Szöveg.
* **meghajtas:**
  + id: A meghajtás típus egyedi azonosítója. Egész szám. Elsődleges kulcs.
  + meghajtas: A meghajtás típus megnevezése. Szöveg.
* **kivitel:**
  + id: A kivitel egyedi azonosítója. Egész szám. Elsődleges kulcs.
  + kivitel: A kivitel megnevezése. Szöveg.
* **special:**
  + id: A különleges ajánlat egyedi azonosítója. Egész szám. Elsődleges kulcs.
  + autoid: Az ajánlathoz tartozó autó azonosítója. Egész szám. Idegen kulcs.
  + text: Az autót bemutató szöveg. Szöveg.
* **favorites:**
  + id: A kedvenc egyedi azonosítója. Egész szám. Elsődleges kulcs.
  + userId: A felhasználó azonosítója, akihez a kedvencek közé sorolt autó tartozik. Egész szám. Idegen kulcs.
  + carId: Az autó azonosítója, amelyet a felhasználó a kedvencek közé rakott. Egész szám. Idegen kulcs.
* **users:**
  + id: A felhasználóhoz tartozó egyedi azonosító. Egész szám. Elsődleges kulcs.
  + username: A felhasználóhoz tartozó felhasználónév. Szöveg.
  + password: A felhasználóhoz tartozó jelszó, melyet titkosítva tárolunk. Szöveg.
  + name: A felhasználó teljes neve. Szöveg.
  + email: A felhasználó email címe. Szöveg.
  + createdate: A felhasználó regisztrációjának dátuma. Automatikusan kerül létrehozásra. Dátum.
  + jog: A felhasználó jogosultsági szintje. Egész szám.
  + hirlevel: A felhasználó feliratkozott-e a hírlevekre. Tinyint.

Relációk:

|  |  |  |
| --- | --- | --- |
| **Idegen kulcs** | **Elsődleges kulcs** | **Kapcsolat típusa** |
| auto.gyartoid | gyarto.id | Egy a többhöz. |
| auto.allapotid | allapot.id |
| auto.szinid | szin.id |
| auto.uzemanyagid | uzemanyag.id |
| auto.valtoid | valto.id |
| auto.meghajtasid | meghajtas.id |
| auto.kivitelid | kivitel.id |
| special.autoid | auto.id |
| favorites.userId | users.id |
| favorites.carId | auto.id |

# Folyamatok

**Autó bemutatása (különlegesek + keresés)**: A felhasználó által kiválasztott autóról betölti az adatbázisban szereplő adatokat. Ezek mellett hozzátudja adni a kedvenceihez. Nem utolsó sorban, az összes autó között tud szűrni akár kombinált feltételekkel.

**Bejelentkezés**: A regisztrált felhasználót felhasználónév-jelszó páros alapján azonosítja. Ezek után megkapja a hozzárendelt jogosultsági szintet, melyet a sessionbe tárolunk.

**Regisztráció**: A felhasználó miután megadta a szükséges adatokat, a rendszer eltárolja az adatait az adatbázisban, a jelszót persze titkosítva.

**Kijelentkezés**: A felhasználó által használt sessiont megszünteti.

**Felhasználó adatainak megjelenítése/módosítása**: A felhasználó meg tudja nézni a saját adatait, sőt azokat tudja módosítani is.

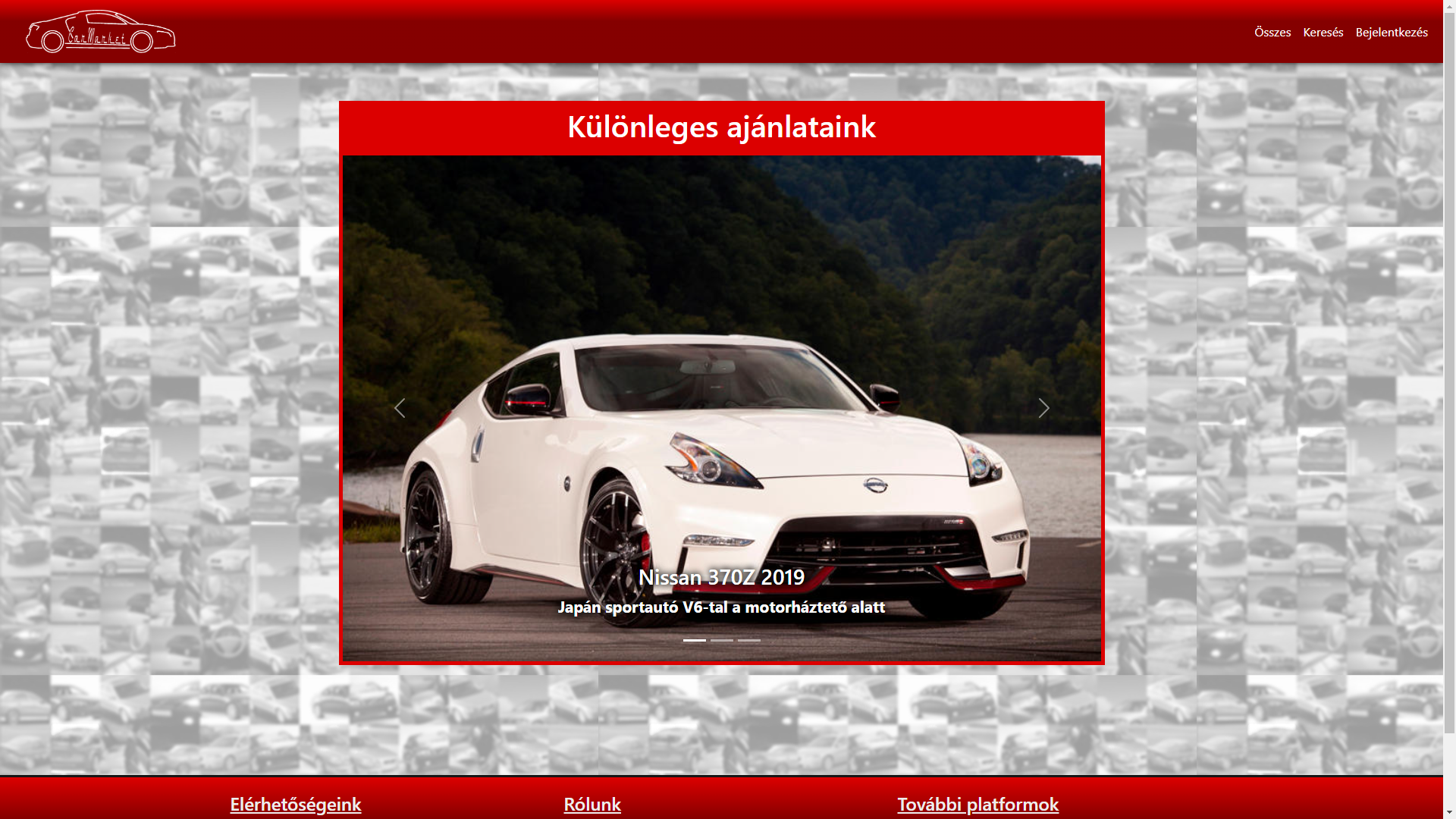
# Felhasználói dokumentáció

## Általános látogató:

A weboldal megnyitásához, először nyissunk meg egy böngészőt (Google Chrome, Mozilla Firefox, Brave stb.) és írjuk be a fejlécbe a következőt: <https://localhost/>. Ezzel meg is nyitottuk a weboldalunk kezdőoldalát.

### Kezdőoldal:

A kezdőoldalon megtalálhatóak a cég aktuális különleges ajánlatai. Ezeket lapozni is lehet a két nyílra kattintással bal és jobb oldalt. Ha rákattint valamelyik autóra, akkor annak megtudja nézni az adatlapját. Emellett a kezdőoldalon megtalálható felül a navigációs sáv.



### Navigációs sáv:

A navigációs sáv az összes oldalon végig kísér. A balszélén megtalálható a cég logója, melyre ha rákattint, mindig a kezdőoldalra tölti vissza. Jobb szélen 3 menü található:

* Összes: Erre kattintva meg tudja tekinteni az autókereskedés összes autóját ABC sorrendbe felsorakoztatva
* Keresés: Erre kattintva megnyitja a keresési oldalt, ahol különböző feltételek alapján tud szűrni a járművekre
* Bejelentkezés: Erre kattintva megnyitja a bejelentkezési ablakot, ahol a felhasználónév-jelszó páros használatával be tud jelentkezni.
* Ha már be van jelentkezve, akkor a „Bejelentkezés” menü helyett, a saját neve fog megjelenni. Ez a menü 3 részre osztódik:
  + Kedvencek: Megnyitja azt az oldalt, ahol meg tudja nézni az összes autót, melyet be rakott a kedvencek közé
  + Adataim: Meg tudja nézni az összes személyes adatát, és módosíthatja is őket
  + Kijelentkezés: Kijelentkezik a fiókjából.





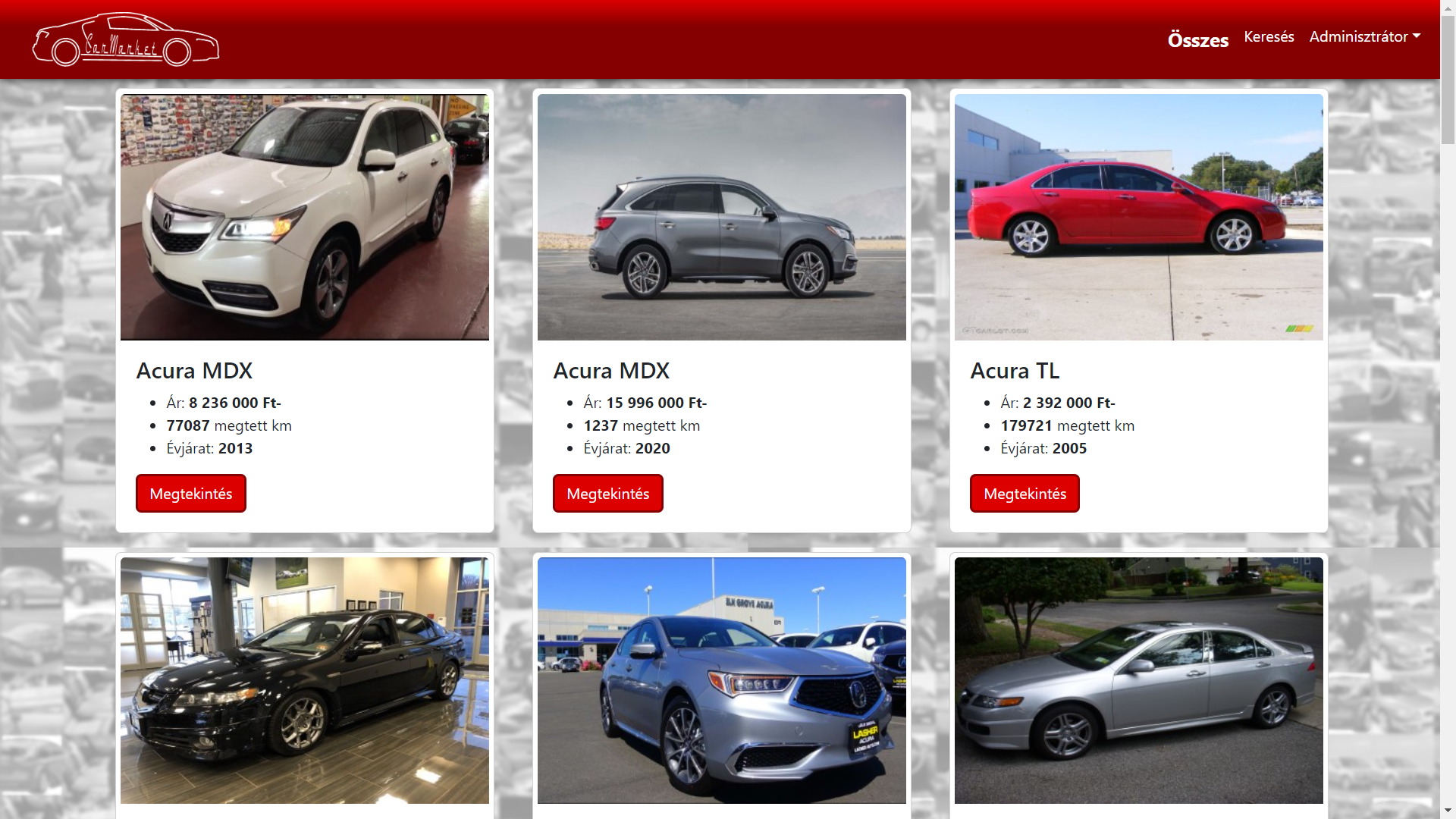
### Lábléc:

A navigációs sáv mellett az oldal alján megtalálható egy **lábléc**, mely ugyanúgy megtalálható minden oldalon. A láblécben különböző információk találhatóak meg a céggel kapcsolatban mint például az elérhetőségek. Továbbá megtalálható a cég többi oldala: Facebook, YouTube, Instagram. Ezek mind linkek, melyre ha rákattint, akkor új ablakban megnyílik az adott weboldal.



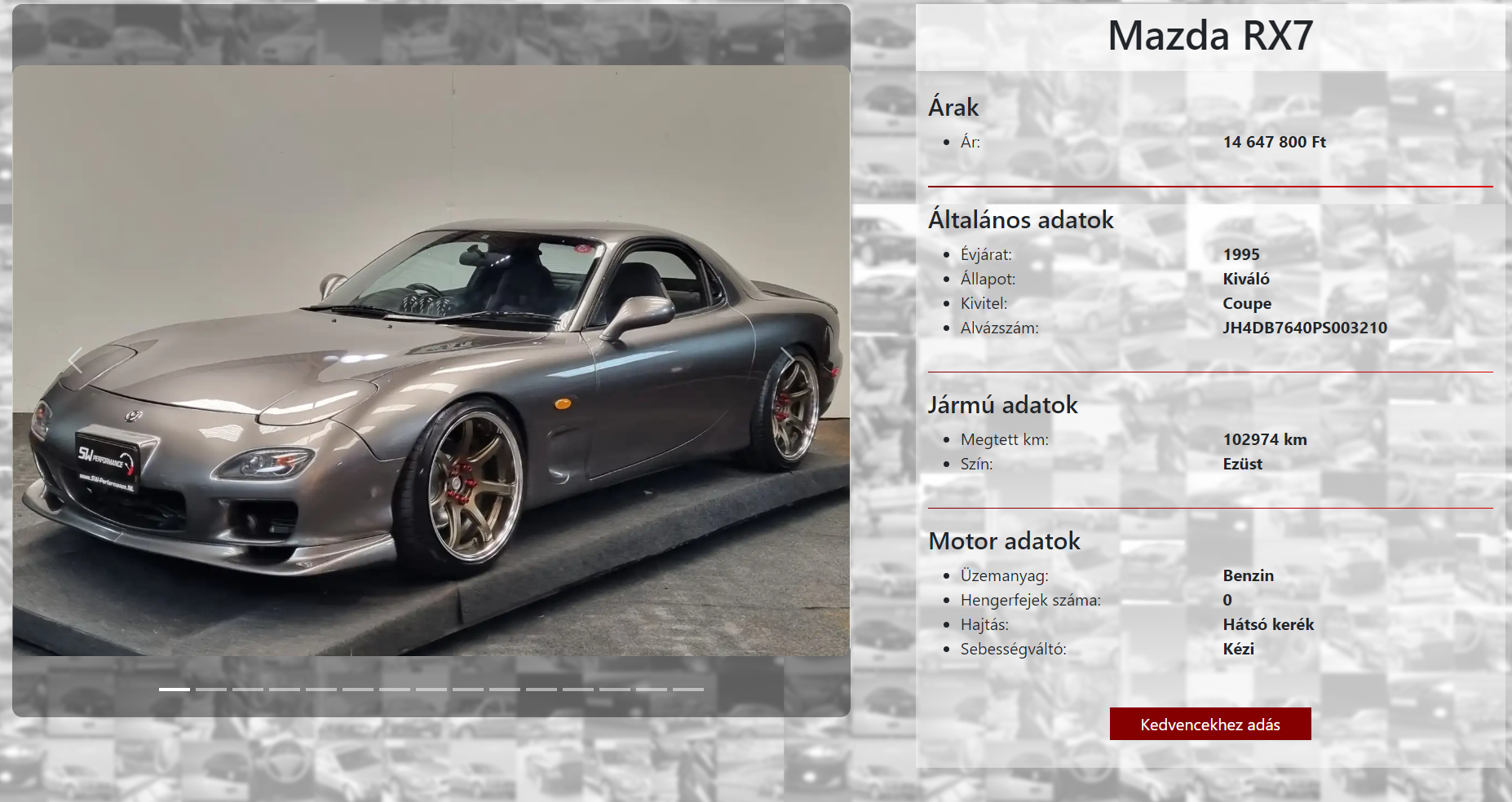
### Összes:

Ha a navigációs sávban rákattint az „Összes” feliratú mezőre, akkor meg tudja tekinteni a cég minden egyes eladó autóját. Az autók sorokba vannak rendezve soronként 3 autóval. Az autók harmincasával töltődnek be, és mindig betölti a következő 30-at, amikor legörget az oldal aljára, addig amíg el nem éri az utolsó autót. Mindegyik autónak van egy saját kártyája, melyben csupán a lényegesebb információk találhatóak meg: gyártó, modell, ár, megtett kilométer, évjárat. Továbbá mindegyik autóról megtalálható egy kép, és a kártya alján van egy „Megtekintés” feliratú gomb. A „Megtekintés” gomb megnyitja az autó adatlapját, ahol további információk találhatóak meg az autókról.



### Autó adatlapja:

Ha rákattintott a „Megtekintés” gombra, akkor megnyílik az autó adatlapja. Ezen az oldalon már sokkal több információ van az autókról, igazából minden. Itt már nem csak egy, hanem több kép is megjelenik a járműről, melyet bal és jobb oldalt lehet lapozgatni, a fehér nyilak segítségével. Jobb oldalt fel vannak sorolva a kocsi tulajdonságai: gyártó, modell, ár, évjárat, állapot, kivitel, alvázszám, megtett kilométer, szín, üzemanyag típus, hengerfejek száma, meghajtás, sebességváltó típusa. Továbbá jobb oldalt lent, megtalálható egy piros „Kedvencekhez adás” feliratú gomb. Ahhoz, hogy a gombnak működjön a funkciója, be kell hogy legyen jelentkezve. Ha nincs bejelentkezve, akkor az oldal átírányítja a bejelentkezési ablakra. Ha be van jelentkezve, akkor viszont az autót sikeresen hozzáadta a kedvencekhez. Ilyenkor megváltozik a gomb felirata: „Mégse kedvenc”. Ha rákattint a „Mégse kedvenc” feliratú gombra, akkor az autót eltávolítja a kedvencei közül, és a gomb felirata újra „Kedvencekhez adás” lesz.





### Keresés:

Ha a navigációs sávban rákattintott a „Keresés” menüre, akkor tud keresni/szűrni az autók között. A keresés több feltételből is állhat, de egyik mező se kötelező:

* Gyártó: Alap esetben „Összes” feliratú, mely azt jelenti, hogy minden fajta gyártó autóit be fogja adni a keresés. Ha szeretni szűrni a gyártóra, akkor kattintson rá az „Összes” feliratú legördülő listára, és válassza ki a kívánt gyártót. Ilyenkor a mező neve megváltozik arra a gyártóra amit kiválasztott.
* Modell: Ez a mező nem elérhető addig, amíg nem választ ki egy gyártót. Amíg a „Gyártó” mező „Összes”-re van állítva, addig ez a mező nem elérhető. Amint kiválaszt egy gyártót, onnantól elérhető lesz a legördülő lista. Ha lenyitja a listát, akkor a kiválasztott gyártó összes elérhető modellje megjelenik. Itt már nem csak egyet, hanem többet is kiválaszthat. Ahhoz, hogy kiválassza a modellt, ahhoz a modell nevétől balra található mezőt kell bejelölnie egy szimpla kattintással. Ha nem választ ki egyet se, akkor az összes modellt be fogja adni a keresés.
* Évjárat: Az autó évjáratát 2 évszám közé állíthatja be. Az oldal megnyitásakor ez úgy van beállítva, hogy betöltsön minden egyes autót. Ha módosítani szeretné, azt két féleképpen teheti meg. Az alsó- és felsőhatárnál is két mező található. Egy csúszka és egy másik mező, mely kiírja a kiválasztott évet. A kurzorral tartsa nyomva a csúszkán található pöttyöt, majd engedje fel, amíg meg nem találja az adott évet, vagy írja be a mezőbe a csúszka alatt a pontos évszámot.
* Ár: Az autó árát 2 összeg közé állíthatja be. Az oldal megnyitásakor ez úgy van beállítva, hogy betöltsön minden egyes autót. Ha módosítani szeretné, azt két féleképpen teheti meg. Az alsó- és felsőhatárnál is két mező található. Egy csúszka és egy másik mező, mely kiírja a kiválasztott árat. A kurzorral tartsa nyomva a csúszkán található pöttyöt, majd engedje fel, amíg meg nem találja az adott összeget, vagy írja be a mezőbe a csúszka alatt a pontos mennyiséget.
* Állapot: Ha lenyitja a listát, akkor megjelenik az összes állapot. Mint a modellnél itt is kiválaszthat többet. Ahhoz, hogy kiválassza az állapotokat, ahhoz az állapot nevétől balra található mezőt kell bejelölnie egy szimpla kattintással. Ha nem választ ki egyet se, akkor az összes állapotot be fogja adni a keresés.
* Szín: Ha lenyitja a listát, akkor megjelenik az összes elérhető szín. Mint a modellnél itt is kiválaszthat többet. Ahhoz, hogy kiválassza a színeket, ahhoz a szín nevétől balra található mezőt kell bejelölnie egy szimpla kattintással. Ha nem választ ki egyet se, akkor az összes színt be fogja adni a keresés.
* Üzemanyag: Ha lenyitja a listát, akkor megjelenik az összes üzemanyag fajta. Mint a modellnél itt is kiválaszthat többet. Ahhoz, hogy kiválassza az állapotokat, ahhoz az állapot nevétől balra található mezőt kell bejelölnie egy szimpla kattintással. Ha nem választ ki egyet se, akkor az összes üzemanyagot be fogja adni a keresés.
* Kivitel: Ha lenyitja a listát, akkor megjelenik az összes elérhető kivitel. Mint a modellnél itt is kiválaszthat többet. Ahhoz, hogy kiválassza a kiviteleket, ahhoz a kivitel nevétől balra található mezőt kell bejelölnie egy szimpla kattintással. Ha nem választ ki egyet se, akkor az összes kivitelt be fogja adni a keresés.
* Váltó: Ha lenyitja a listát, akkor megjelenik az összes elérhető váltó típus. Mint a modellnél itt is kiválaszthat többet. Ahhoz, hogy kiválassza a váltókat, ahhoz a váltó típusának a nevétől balra található mezőt kell bejelölnie egy szimpla kattintással. Ha nem választ ki egyet se, akkor az összes váltó típussal rendelkező autót be fogja adni a keresés.
* Meghajtás: Ha lenyitja a listát, akkor megjelenik az összes elérhető meghajtás. Mint a modellnél itt is kiválaszthat többet. Ahhoz, hogy kiválassza a meghajtásokat, ahhoz a meghajtás nevétől balra található mezőt kell bejelölnie egy szimpla kattintással. Ha nem választ ki egyet se, akkor az összes meghajtást be fogja adni a keresés.

Ha kiválasztott minden feltételt amit szeretne, akkor nyomjon rá a „Keresés” feliratú gombra. Amint rákattintott, a lap alján meg fog jelenni az összes autó, mely megfelel a beállított feltételeknek. Az autók ugyanúgy jelennek meg, mint az „Összes” menüben.



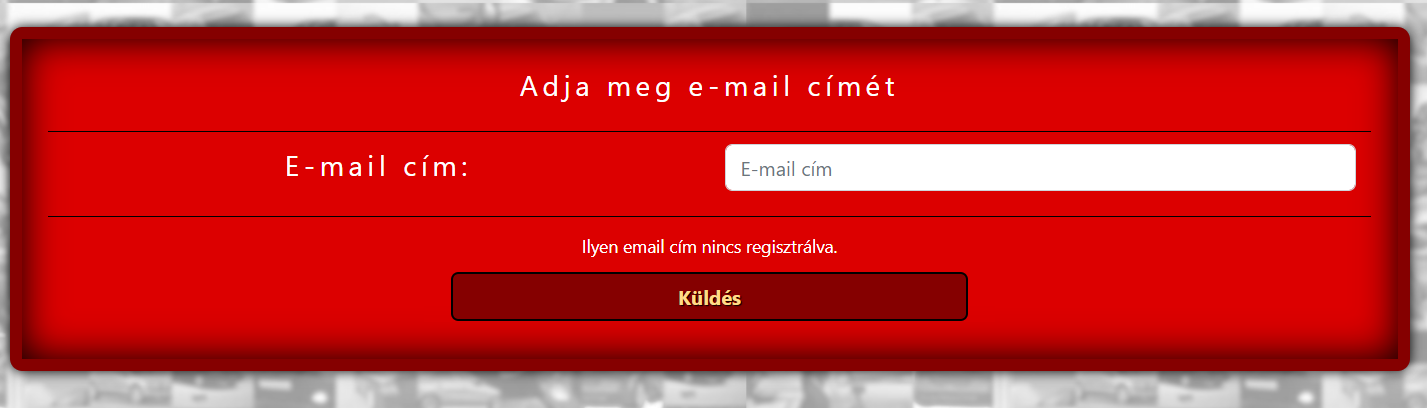
### Bejelentkezés:

Ha a navigációs sávban rákattint a „Bejelentkezés” feliratú gombra, akkor megnyílik az az ablak, ahol be tud jelentkezni. 2 mező van: „Felhasználónév” és „Jelszó”. A mezőkbe írja be a felhasználónevét, és a jelszavát, majd kattintson a „Bejelentkezés” feliratú gombra. Ha még nincs felhasználói fiókja, kattintson a „Még nem regisztrált?” feliratú linkre.



### Elfelejtett jelszó:

Egy hibás bejelentkezés után megjelenik egy hivatkozás a bejelentkezési fülön ezzel a szöveggel: „Elfelejtette a jelszavát?”. Erre a gombra kattintva át lesz irányítva egy másik felületre, ahol a fiókhoz tartozó email cím megadása után a „Küldés” gombra kattintva a megadott email címre kiküldünk egy újonnan generált jelszót. Olyan email cím megadása esetén amely nincs regisztrálva, vissza kap egy hibaüzenetet miszerint „Ilyen email cím nincs regisztrálva.”.





### Regisztráció:

A regisztrációs ablakban több mező is található:

* „Vezetéknév”: Írja be a vezetéknevét. Ezt a mezőt kötelező kitölteni.
* „Keresztnév”: Írja be a keresztnevét. Ezt a mezőt kötelező kitölteni.
* „Email cím”: Írja be az email címét. Ezt a mezőt kötelező kitölteni.
* „Felhasználónév”: Találjon ki magának egy felhasználónevet, majd írja be. Ezt a mezőt kötelező kitölteni.
* „Jelszó”: Találjon ki magának egy jelszót majd írja be. A jelszónak minimum 8 karakter hosszúnak kell lennie. Ezt a mezőt kötelező kitölteni.
* „Jelszó mégegyszer”: Írja be mégegyszer a kitalált jelszót. A regisztráció csak akkor lesz sikeres, ha a két jelszó megegyezik. Ezt a mezőt kötelező kitölteni.
* „Elolvastam és elfogadom az ÁSZF-et és az Adatvédelmi tájékoztatót”: Ennek a mezőnek a kipipálása kötelező. Ezzel elfogadja az Általános Szerződési Feltételeket, és az Adatvédelmi tájékoztatót.
* „Hírlevél feliratkozás”: Ezt a mezőt akkor pipálja ki, ha szeretne az email fiókjára hírleveleket kapni az oldalról. (A hírlevelek küldése egy későbbi funkció lesz a weboldalon).

Ha kitöltött minden mezőt, akkor nyomja meg a „Regisztráció” gombot. A regisztráció csak akkor lesz sikeres, ha sem a felhasználónév, se az email cím nincs még használatban más felhasználó által, és a két beírt jelszó megegyezik és minimum 8 karakter hosszúak. Ezek után az oldal sikeresen bejelentkezteti, és elérheti az új funkciókat.

Ha már van fiókja, és szeretne bejelentkezni, akkor nyomjon az űrlap alján található „Már regisztrált” című linkre, ez átirányítja a bejelentkezési oldalra.

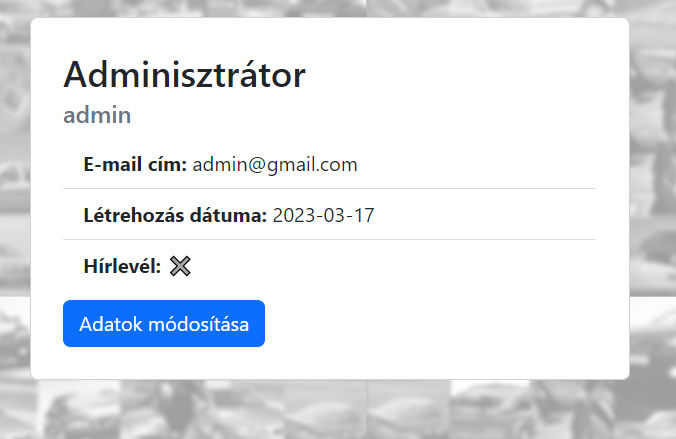


### Kedvencek:

Ez az almenü csak akkor érhető el, ha már be van jelentkezve. Itt megjelenik az összes autó amit hozzáadott a kedvenceihez, ugyanúgy mint az „Összes” menüben.

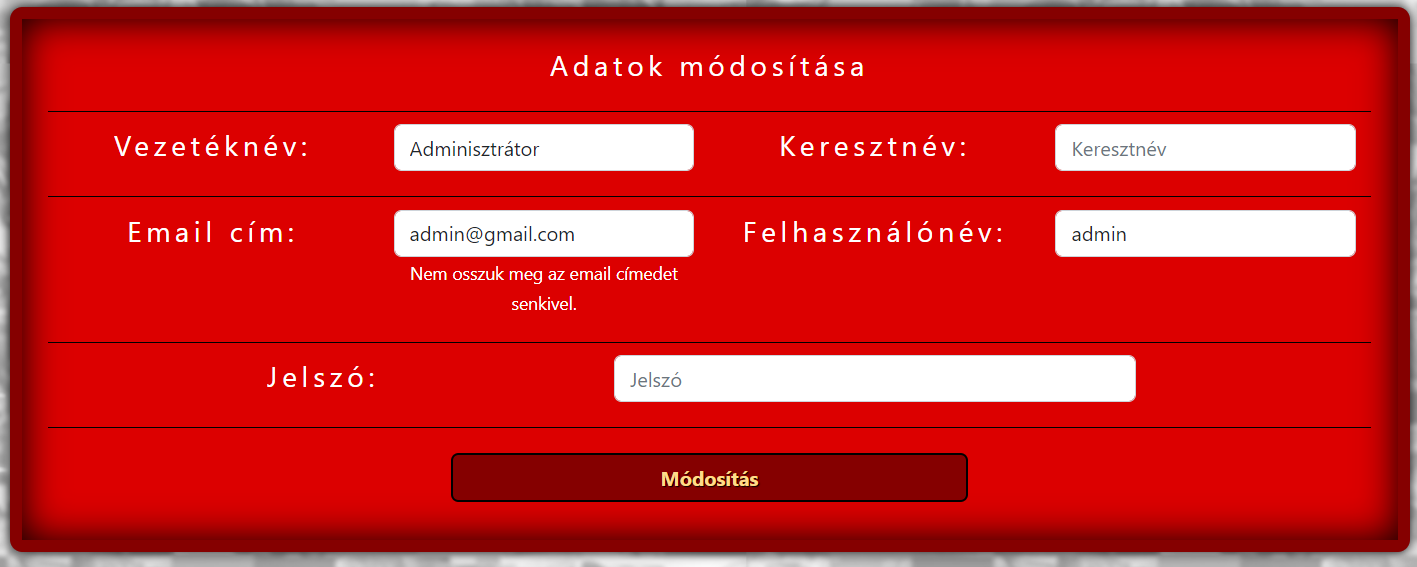
### Adataim:

Ez az almenü csak akkor érhető el, ha már be van jelentkezve. Itt megtalálhat a felhasználói fiókhoz tartozó minden információt (név, felhasználónév, email cím, fiók létrehozásának dátuma, hírlevélre való feliratkozás). Ezeket az adatokat az „Adatok módosítása” gombra kattintva megváltoztathatja.



### Adatok módosítása:

Ha rákattintott az „Adatok módosítása” gombra, akkor megjelenik egy űrlap, ahol a következő adatait módosíthatja: vezetéknév, keresztnév, email cím, felhasználónév. A módosításhoz meg kell adnia jelszavát. Ha beírta a megfelelő adatokat, nyomja meg a „Módosítás” gombot, és ha jól adta meg a jelszót, és az email cím és a felhasználónév nem foglalt, akkor sikeres az adat módosítás.



### Kijelentkezés:

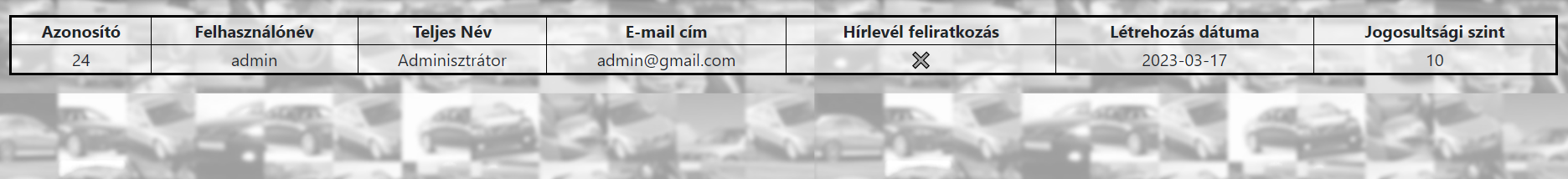
Erre a gombra kattintva, kijelentkezik a felhasználó fiókjából.

## Adminisztrátor:

Ha adminisztrátorként van bejelentkezve, akkor egy új almenü jelenik meg az „Adataim” és a „Kijelentkezés” között: „Felhasználói fiókok”.

### Felhasználói fiókok:

Erre a menüre kattintva, meg tudja nézni az összes regisztrált felhasználó adatát. Az információk közül egyedül a jelszó hiányzik, mivel az titkosítva van eltárolva.



# Források

## Adatbázis:

Az adatbázis alapját a <kaggle.com>-ról töltöttem le, amit a saját szükségleteimre módosítottam.

## Fogalmak:

HTML5: <https://hu.wikipedia.org/wiki/HTML>

express: <https://expressjs.com/>

express-session: <https://www.section.io/engineering-education/session-management-in-nodejs-using-expressjs-and-express-session/>

cookie-parser: <https://www.tutorialspoint.com/expressjs/expressjs_cookies.htm>

ejs: <https://ejs.co/>

morgan: <https://www.npmjs.com/package/morgan>

MySQL: <https://hu.wikipedia.org/wiki/MySQL>

## Képek:

Az összes képet a google képkeresőjéből töltöttem le.

## Programrészletek:

Az asztali alkalmazásban megtalálható egy külön osztály a nyomtatásra. Ezt az osztályt a <https://www.codeproject.com/Articles/18042/Another-DataGridView-Printer> oldalról töltöttem le, és onnan vettem a mintát a használatához.