

SZOFTVERFEJLESZTÉSI ÉS KIVITELEZÉSI DOKUMENTÁCIÓ

Szoftverfejlesztő és -tesztelő technikus
5 0613 12 03

Készítette:
Boros Dániel, Barta János

2/14EC



Debreceni SZC Mechwart András Gépipari és Informatikai Technikum

Debrecen
2024.

Tartalomjegyzék

Bevezetés	3
Követelmény	3
Témaválasztás	3
A téma bemutatása	3
Fejlesztői dokumentáció	5
I. A fejlesztői környezet bemutatása	5
<i>Hardver</i>	5
<i>Szoftver</i>	5
II. A fejlesztés ütemezése	5
III. A szoftver funkciói	6
<i>Szerver oldali funkciók</i>	6
<i>Kliens oldali funkciók</i>	6
IV. Érdekesebb algoritmusok, funkciók	6
V. Tesztelés	7
VI. Adatbázis	8
VII. Továbbfejlesztési lehetőségek	9
Felhasználói dokumentáció	10
I. Telepítés és indítás	10
II. Belépés	11
III. Szűrés és kiválasztás	12
IV. Statisztika	13
Összefoglalás	15
Irodalomjegyzék	16

Bevezetés

Követelmény

A szoftverfejlesztő és -tesztelő szakmai képzés során a szakmai vizsga megkezdésének feltétele a vizsgaremek elkészítése, és a megfelelő formában történő leadása.

A projektmunkánk során elkészített vizsgaremeknek az alábbi elvárásoknak kell megfelelnie:

- Életszerű, valódi problémára kell megoldást nyújtania.
- Adattárolási és -kezelési funkciókat is meg kell valósítania.
- RESTful architektúrának megfelelő szerver és kliens oldali komponenseket kell tartalmaznia.
- A kliens oldali komponensnek alkalmasnak kell lenni asztali és mobil eszközön történő használatra.

Témaválasztás

A követelmények ismeretében választásunk egy weboldal elkészítésére esett, amely frontend és backend részből épül fel, segítségével adatok tárolása, lekérdezése, bemutatása valósítható meg.

Célkitűzésünk az volt, hogy a képzés során megszerzett ismeretek birtokában egy hasznos és látványos alkalmazást hozzunk létre.

A téma valós helyzetre próbál reflektálni. Az összetett nyilvántartási és ütemezési feladat megoldása túlmutat ezen képzés keretein, így a honlap továbbfejlesztési lehetőségeit a későbbiekben részletesen bemutatjuk.

A téma bemutatása

Vannak olyan szervezetek, vállalkozások, ahol elengedhetetlen a mobilitás. Ez elsősorban gépjárművekkel oldható meg. Egy ponton minden vállalkozásban felmerül az igény, hogy járműveikkel kapcsolatos információkat egy helyen kezeljék, nagyobb gépjármű állománynál is bármikor lekérdezhető legyen, hogy ezek az eszközök a feladat elvégzésére rendelkezésre állnak-e.

Létre kívántunk hozni egy felületet, ahol a szervezet a saját gépjármű állományáról, a gépjárművek használatára jogosult felhasználók adatairól tárol információkat.

A szervezet gépjármű ügyintézésével megbízott munkatársa rögzíti a gépjármű adatokat, valamint a felhasználói adatokat.

A felhasználók a kapott felhasználónévvel jelszóval igénylésüket a flottakezelő rendszer felületén keresztül nyújthatják be. Ott láthatják, hogy adott időpontra mely gépjárművek érhetőek el.

A munkatársak részére Basic, Comfort, valamint Prémium kategóriájú járművek állnak rendelkezésre - beosztásuktól, jogosultsági szintjüktől függően.

Tehát ez a felület lehetőséget biztosít a munkatársak részére az elérhető járműveket munkájukhoz megigényelni, lefoglalni.

A fentieket egy CarManagement elnevezésű weboldal formájában valósítottuk meg, amit a felhasználóknak böngészőn keresztül lehet elérni.

Fejlesztői dokumentáció

I. A fejlesztői környezet bemutatása

Hardver

A fejlesztéshez szükséges számítógépnek alkalmasnak kellett lenni a virtuális környezet futtatására. Ez a mai hardverekkel, ami a Windows 10 operációs rendszert is tudja futtatni, nem okozott problémát. A fejlesztés során aktív internetkapcsolat is szükséges volt, ami segített a fejlesztői munkával járó keresésekben, valamint a projekt előrehaladtával a verziók egymás közti megosztásában.

Szoftver

A megvalósítás frontend oldalon HTML5 oldalszerkezet kialakításával, strukturális elemeinek attribútumainak alkalmazásával, stílusok definiálásával és alkalmazásával, Bootstrap keretrendszer használatával (tipográfiai elemek, konténerek, reszponzív viselkedést biztosító rácsok, szövegek elrendezése, listák, táblázatok formázása, képek kezelése, tartalom elkülönítése, panelek formázása, gombok kialakítása) történt.

Javascript függvények használatával a végpontok által szolgáltatott adatokból dinamikus weboldal került kialakításra, ami az előre definiált struktúrák alkalmazásával szűrési lehetőséget biztosít a felhasználó számára.

A backend szerver oldalt egy nyílt forráskódú Python nyelven alapuló Django webes keretrendszerrel, a szükséges adatbázis táblákat SQLite táblákkal oldottuk meg.

A járművek képei a márkakereskedői oldalakról származnak, a logó saját tervezés, valamint a háttérkép saját fotó.

Mindezekhez a Visual Studio Code (forráskód írás, szerkesztés, verziókezelés), GitHub (verzió kezelés), Figma (oldalterv), Inkscape (logó megtervezése), PowerPoint, Fényképek alkalmazás (képszerkesztés, konvertálás), Bootstrap5 (CSS keretrendszer), Django szerver oldali keretrendszer használatára volt szükség.

A megjelenítésre, a tesztelésre, a reszponzivitás ellenőrzésére és a hibakeresésre a Google Chrome és a Microsoft Edge böngészőket használtuk. A lokális szerverrel történő fejlesztés miatt szükséges volt a böngésző egy bővítményét is telepíteni (Allow CORS: Access-Control-Allow-origin).

II. A fejlesztés ütemezése

A fejlesztést több ütemben valósítottuk meg:

1. A projektötlet néhány oldalas kidolgozása
2. Frontend kezdeti vázlat (Figma online)

3. Backend - adatbázis létrehozása (Django)
4. Frontend menürendszer, mintaoldalak készítése (HTML5, CSS, Bootstrap5)
5. Dinamikus oldal létrehozása (JavaScript, Django)
6. Bejelentkezési lehetőség megoldása (HTML5, Bootstrap5, Django)
7. A szerver oldali frontend szolgáltatás kialakítása (Django)
8. Végző funkcionális tesztelés (Microsoft Edge, Google Chrome)

III. A szoftver funkciói

Szerver oldali funkciók

A Django szerver oldali környezet adminisztrációs felületén bejelentkezés után lehetőség van az adminisztrátornak az alkalmazott adatbázisok feltöltésére, szerkesztésére. Így a járművek, a felhasználók és a foglalási időpontok adatai itt szerkeszthetők.

Kliens oldali funkciók

Bejelentkezés, a felhasználó azonosítása. API végpontok alapján lekérdezés, szűrési lehetőségek (felhasználó jogosultsági kategória, jármű kategória, márka).

Amit szerettünk volna még a projekt során megvalósítani, és csak részben valósult meg:

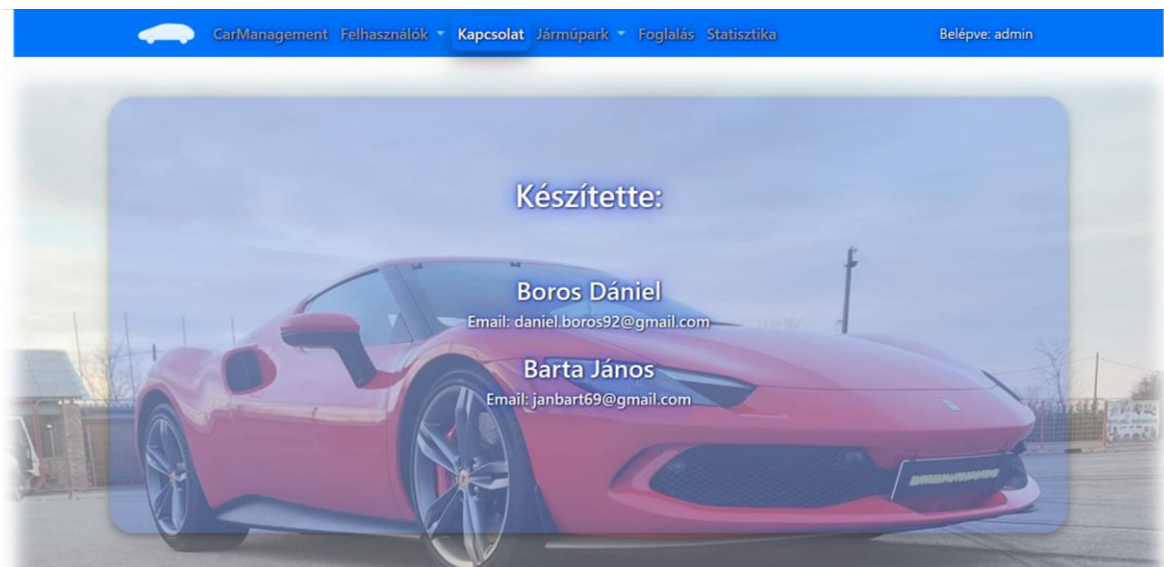
- *A felhasználók regisztrációja kliens oldalról.*
A regisztrációs oldal, valamint a backend rész is elkészült. Az előzetes tesztelés során („live server”) az ellenőrző funkciók is működtek (pl. felhasználó névnel karakter minimális száma), azonban az adatbázishoz történő kapcsolódás során hibába ütköztünk, amit idő hiányában már nem tudtunk kiküszöbölni. Ez a rész egyelőre kikerült a projektből. A regisztráció jelenleg az adminisztrációs felületen érhető el.
- *A foglalási adatok megadása kliens oldalról.*
Az egyes járművek adatlapja tartalmazza a foglalási naptárat. A statikus adatlap minta a naptárral elkészült. A foglalási adatok jelenleg a backend részben rögzíthetők.
A kiválasztott jármű adatlapjának és a bejelentkezett felhasználó általi járműfoglalás dinamikussá tételére kellő idő hiányában már nem került sor.

IV. Érdekesebb algoritmusok, funkciók

A projekt megvalósítása során érdekességképpen kiemelnénk az adatbázisok, modellek kialakítását, szűrési lehetőségek kidolgozását.



Kiemelnénk még az autentikáció folyamatát, a hibás belépési azonosító kezelését, a belépés nélküli oldalak megtekinthetőségének korlátozását. Továbbá az árnyékolások használatát, ezzel látványosabb megjelenést biztosítva weboldalunknak.



V. Tesztelés

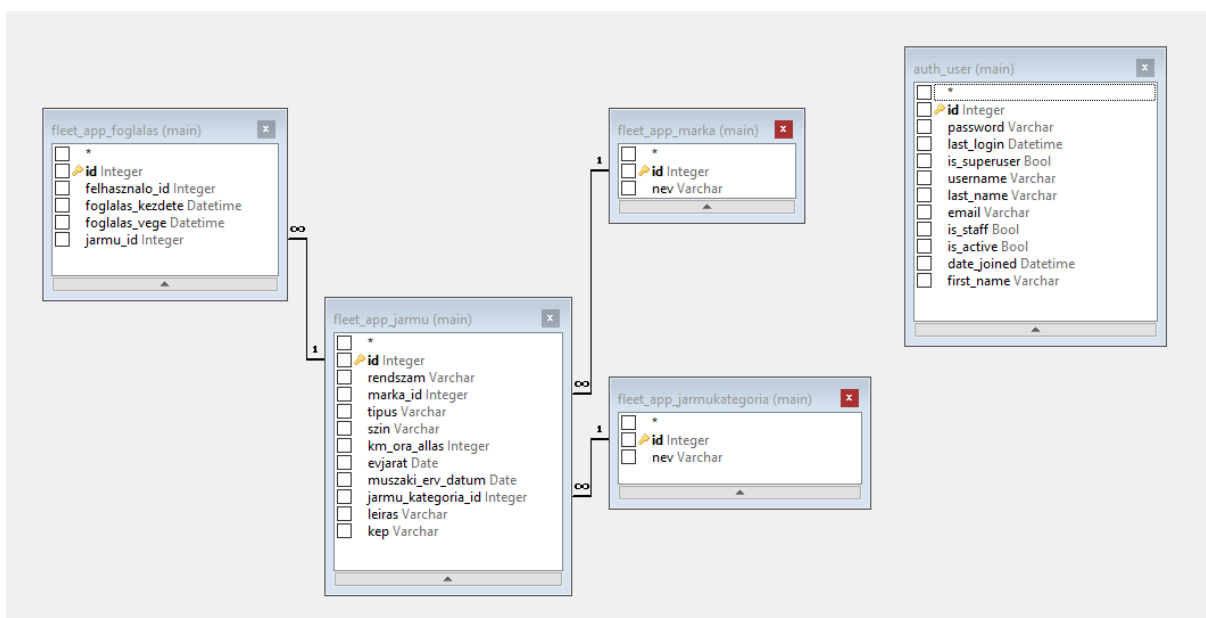
Folyamatos manuális szoftverteszteket végeztünk, így funkcionális tesztelést melynek során ellenőriztük, hogy az alkalmazás megfelel-e a funkcionális követelményeknek. Integrációs teszteléssel győződünk meg arról, hogy az egyes komponensek együttműködése biztosított-e. Az alkalmazás felhasználói élményét folytonosan teszteltük (UX tesztelés). Kompatibilitási teszteléssel ellenőriztük, hogy az alkalmazás különböző környezetekben (pl. eltérő böngészőkben) és eszközökön (fejlesztői gépek, valamint iskolai gépek) megfelelően működik-e. A responzív viselkedés tesztelését a böngésző felület átméretezésével (a DevTools-ban eszközök szimulációjával is) végeztük.

Ellenőriztük, hogy a legújabb változtatások nem okoztak-e hibákat a korábban működő részekben (regressziós tesztelés).

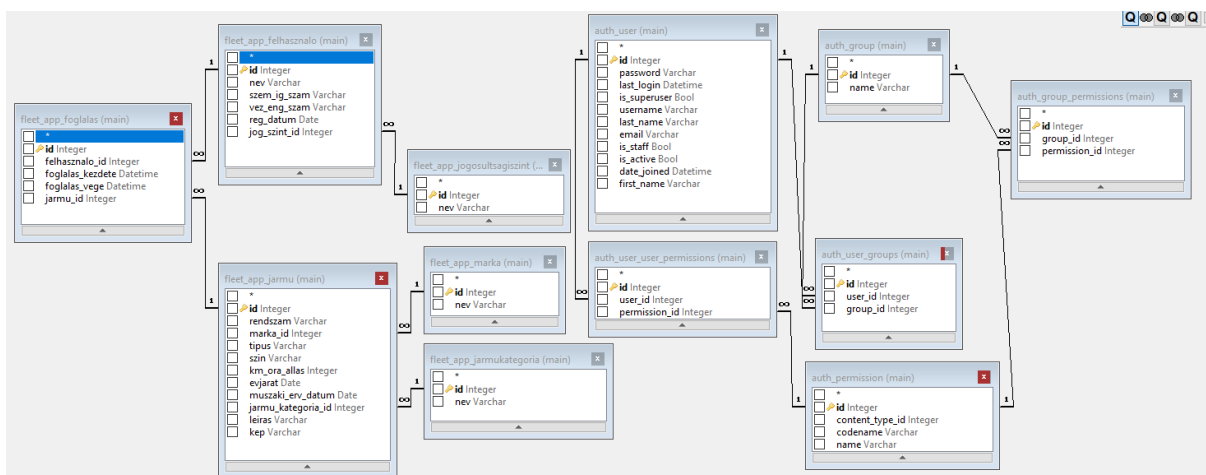
Szükség esetén az adott egységeket (unit) fejlesztés közben folyamatosan teszteltük. A sikertelen tesztek eredményeképp, vagy nem az elvárt működést produkálva az adott elemek javításra, vagy esetleg átütemezésre (pl. a kliens oldali regisztráció) kerültek.

VI. Adatbázis

A projekthez kapcsolódó SQLite adatbázist az alábbi ábra szemlélteti. A kép bemutatja a Django szervert oldali keretrendszerben megtalálható „auth_user” táblát, valamint a projektben létrehozott „fleet_app_foglalas”, „fleet_app_jarmu”, „fleet_app_jarmukategoria” és „fleet_app_marka” táblákat a mezőnevekkel, valamint a táblák közötti kapcsolatokkal. A fejlesztés jelenlegi szakaszában ezeket a táblákat használjuk.



A továbbfejlesztés során azonban több táblára is szükség lesz. Ezért az alábbi képen a jogosultsági szinttel kiegészített táblák, valamint a Django beépített táblái látszanak.



VII. Továbbfejlesztési lehetőségek

A projekt a jelenlegi állapotán jelentősen túlmutató potenciállal rendelkezik. Ezeket pontokba szedve, időrendi ütemezésben az alábbiakban foglaltuk össze:

- Reszponzív viselkedés további tesztelése, korrigálása. Így a menürendszer telefonokra történő átváltása („hamburgermenü”).
- A kliens oldali bejelentkezés megoldása. A jármű oldal és a foglalások dinamikussá tétele, jogosultsági szintekkel kiegészítve.
- A járművek, felhasználók adminisztrációs felületének kliens oldalra helyezése.
- Statisztikai lekérdezési lehetőség integrálása, az oldal dinamikussá tétele.
- Lokális szerver helyett vállalati Intraneten, vagy Interneten keresztüli elérés kialakítása.
- Külső szerverről elérhető több vállalati felhasználó részére nyújtott szolgáltatás, megfelelő biztonsági és jogosultsági szintekkel kialakítva.
- Autókölcsönzéssel foglalkozó vállalkozás részére is megfelelő felület kialakítása. Kártyás fizetés integrálása.

I. Telepítés és indítás

Az alkalmazás minimális hardverigénye nem haladja meg az operációs rendszer hardverigényt, tehát amennyiben az adott operációs rendszer futtatható a számítógépen úgy az alkalmazás is.

Mivel weboldalról beszélünk, használatához egy webböngészőre van szükség. Az alkalmazás lokális szerverre készült, ezért a megfelelő futtatáshoz, a kiszolgáló backend keretrendszer telepítése szükséges.

A Django-hoz telepített Python programnyelv szükséges, ezért annak ellenőrzéséhez, hogy a rendszeren telepítve van-e a, futtassa ezt a parancsot a parancssorban:

```
python --version
```

Ha a Python telepítve van, akkor a verziószámmal megegyező eredményt kapunk. Amennyiben nincs telepítve a Python, ingyenesen letölthető a következő weboldalról: <https://www.python.org/>

A Django telepítéséhez olyan csomagkezelőt kell használni, mint a PIP, amely a Python 3.4-es verziójától található meg.

Annak ellenőrzéséhez, hogy a rendszeren telepítve van-e a PIP, futtassa ezt a parancsot a parancssorban:

```
pip --version
```

Ha a PIP telepítve van, egy eredményt kapunk a verziószámmal.

Amennyiben ezzel elkészültünk jöhet a virtuális környezet megteremtése, valamint a projekt bemásolása a számítógépre. Ennek menete lépésenként az alábbiakban található.

A parancsokat Parancssorban kell kiadni, mivel a PowerShell-ben le van tiltva a scriptek futtatása!

1. Start gombra kattintva beírjuk, hogy „CMD”, ezután egy Enter gombbal elindítjuk a parancssori ablakot.
2. Carman mappa létrehozása tetszőleges helyen

```
mkdir carman
```

3. Belépés a carman mappába

```
cd carman
```

4. Python virtuális környezet (venv) létrehozása a carman mappában

```
python -m venv env
```

5. Belépés a virtuális környezet mappájába

```
cd env
```

6. Virtuális környezet aktiválása (ha megjelenik a (env) a konzolon akkor működött)

```
Scripts\activate
```

7. Visszalépés a carman mappába az env mappából

```
cd..
```

8. Projekt repository klónozása a lokális repositoryba

```
git clone https://github.com/brsdnl92/carman.git
```

9. Belépés a belső carman mappába

```
cd carman
```

10. Belépés a backend mappába

```
cd backend
```

11. Python csomag követelmények telepítése

```
pip install -r requirements.txt
```

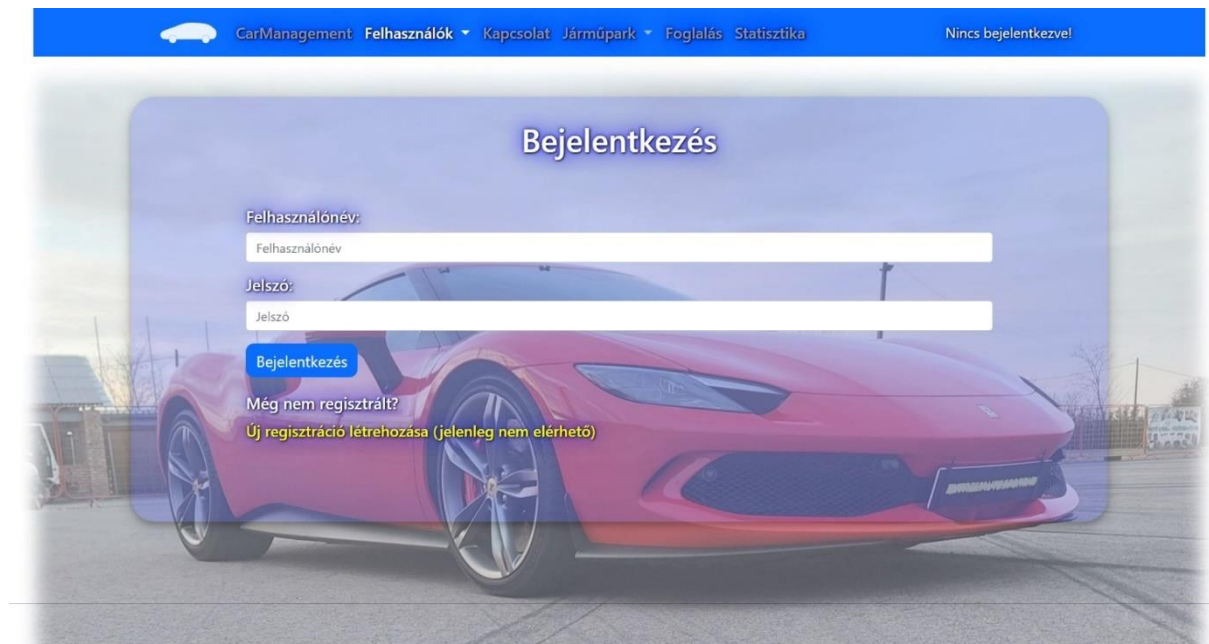
12. Backend localhost szerver futtatása (http://127.0.0.1:8000/)

```
python manage.py runserver
```

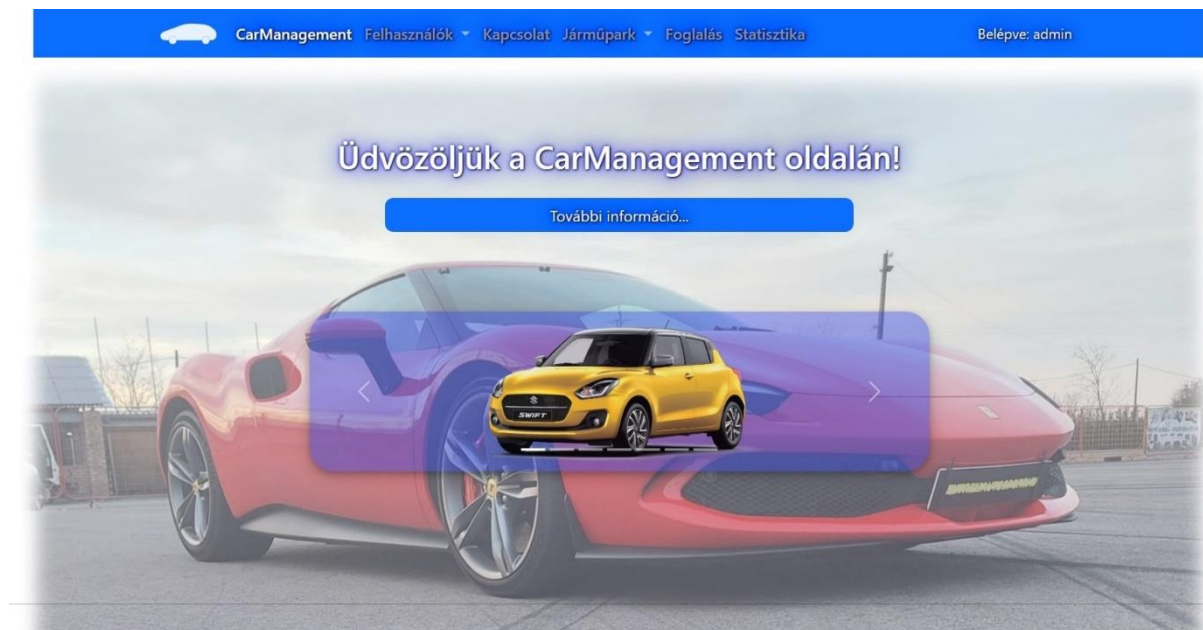
Ha jól dolgoztunk, akkor a böngészőbe bemásolva a localhost IP címét: <http://127.0.0.1:8000/> a bejelentkezési felületre jutunk.

II. Belépés

Alapesetben az 'admin' felhasználónevet, illetve jelszót kell megadni a bejelentkezéshez. Míg nem lépünk be, addig nem érhető el minden oldal.



Bejelentkezés után a kezdő oldalra jutunk.

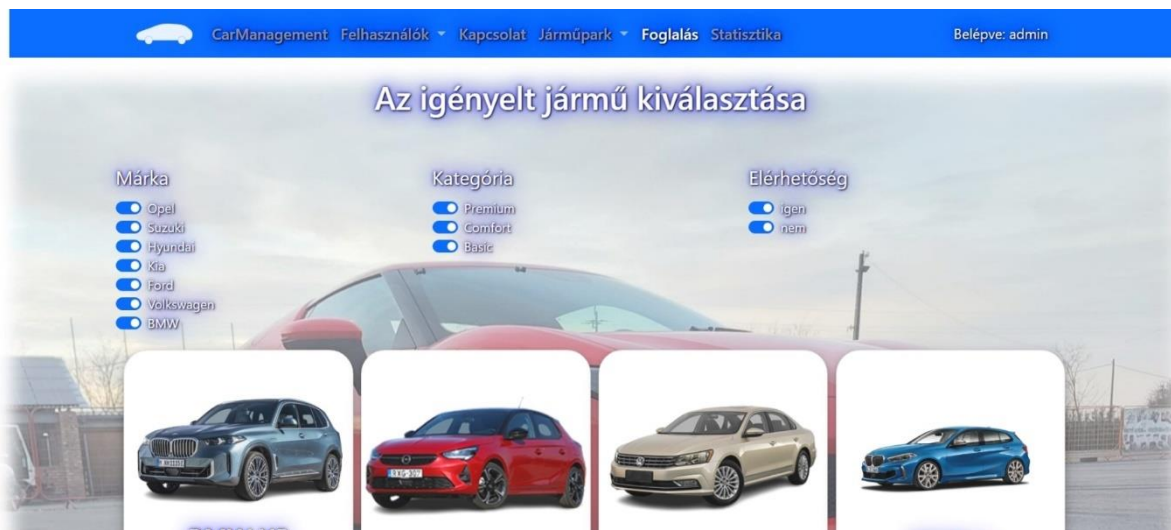


A backend felületre a <http://127.0.0.1:8000/admin> oldalon jelentkezhetünk be a fent említett felhasználónév-jelszó párossal. Itt lesz lehetőség hozzáadni, szerkeszteni új járműveket, illetve új felhasználókat.

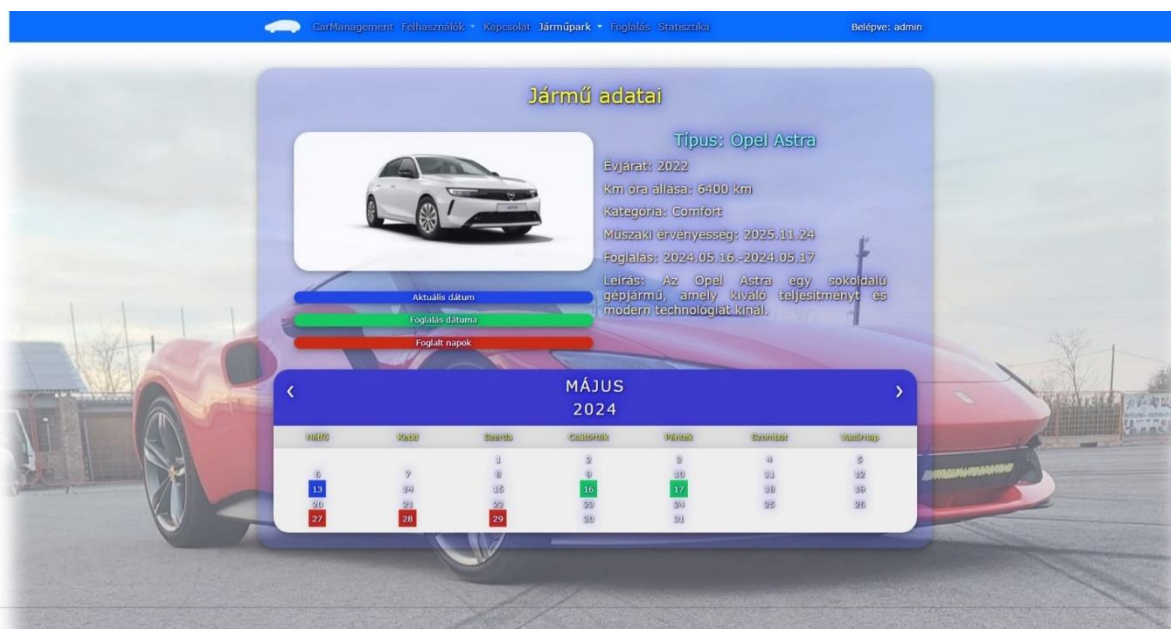
The image shows the login interface of the Django administration system. It has a dark blue header with the text 'Django administration' and a small circular icon. Below the header, there are two input fields: one for 'Username:' and one for 'Password:'. At the bottom of the form is a blue button labeled 'Log in'.

III. Szűrés és kiválasztás

Az alkalmazásban lehetőség van gépjármű márka, kategória és elérhetőség szerinti szűrést végezni.

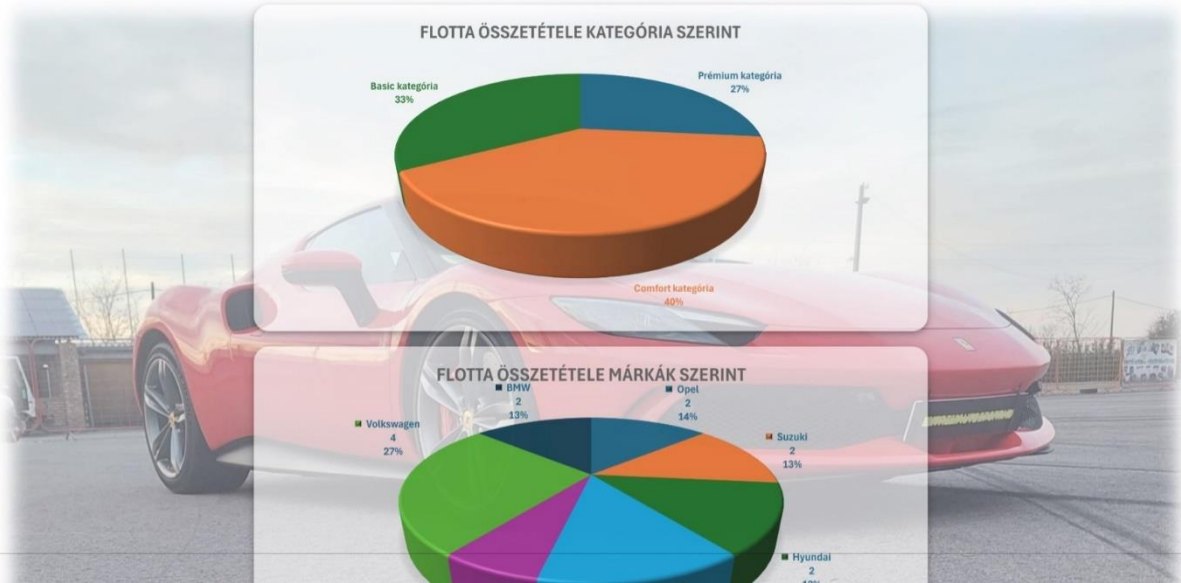


Ezt követően kiválasztható az adott gépjármű adatlapja, ahol lehetőség nyílik a foglalások megtekintésére, valamint rögzítésére. (Ez utóbbi oldalnak idő hiányában csak a statikus sablonja készült el. Így a kiválasztás is csak erre az oldalra mutat.)



IV. Statisztika

Terveink szerint a rögzített adatok alapján statisztikákat lehet majd lekérdezni egy dinamikus weblapon keresztül. Ennek kezdeti lépését egy statikus oldal formájában a projektbe beillesztettük.



Összefoglalás

Az alkalmazás dokumentációja bemutatja a szoftverfejlesztő és -tesztelő technikus képzés végeztével beadandó vizsgaremeket, ami jelen esetben egy weboldal formájában valósult meg.

A projekt célja az elsajátított frontend és backend ismeretek alkalmazása egy gyakorlati probléma megoldása kapcsán.

A dokumentáció részletezi a fejlesztői környezetet, a felhasznált szoftvereket. Felvázolja a megvalósítás lépéseit, a továbbfejlesztési lehetőségeit. Bemutatja a felhasználó szemszögéből a használatba vétel lépéseit, továbbá a főbb funkciókat.

Végül köszönetet szeretnénk mondani a Debreceni SZC Mechwart András Gépipari és Informatikai Technikum tanárainak a lelkiismeretes munkájukért, segítségükért, türelmükért.

Irodalomjegyzék

- Követelményrendszer:
https://api.ikk.hu/storage/uploads/files/5_0613_12_03_szoftverfejleszto_es_tesztelo_azonnali_november_ala-1701347102653.pdf
- Python: <https://www.python.org>
- Django: <https://www.djangoproject.com>
- HTML, JavaScript, CSS, Bootstrap, Python, Django:
<https://www.w3schools.com>