**Debreceni SZC Baross Gábor Technikum, Szakképző Iskola és Kollégium**

**Autó kölcsönző**

**Készítette:  
Juhász Tamás**

**Tartalomjegyzék:**

[**1. Bevezetés:** 3](#_Toc131320872)

[Témaválasztás indoklása: 3](#_Toc131320873)

[Elméleti és gyakorlati jelentőség: 3](#_Toc131320874)

[**2. Fejlesztői dokumentáció:** 3](#_Toc131320875)

[Weblapon használt adatszerkezetek: 4](#_Toc131320876)

[Az adatbázis egyedi táblái és azok mezői: 4](#_Toc131320877)

[A user tábla: 5](#_Toc131320878)

[A termékek tábla: 5](#_Toc131320879)

[A kosár tábla: 6](#_Toc131320880)

[**3 .Egyedi algoritmusok az oldalon:** 6](#_Toc131320881)

[A regisztráció algoritmusa: 6](#_Toc131320882)

[A beléptetés algoritmusa: 7](#_Toc131320883)

[A kijelentkezés algoritmusa: 8](#_Toc131320884)

[A végösszeg számlálás algoritmusa: 8](#_Toc131320885)

[**4. JavaScript funkciók:** 9](#_Toc131320886)

[Regisztráció funkció a regisztráció gombhoz: 9](#_Toc131320887)

[Felhasználó és rang kiírása: 11](#_Toc131320888)

[Navigációs fül: 11](#_Toc131320889)

[Kosárba helyezés: 11](#_Toc131320890)

[**5. Teszt dokumentáció:** 14](#_Toc131320891)

[Az oldal tesztelése és a felmerülő hibák: 14](#_Toc131320892)

[**6. Fejlesztési lehetőségek:** 14](#_Toc131320893)

[**7. Webes felület célja:** 15](#_Toc131320894)

[**8. Hardware és szoftver igény:** 15](#_Toc131320895)

[Telepítés és indítás: 15](#_Toc131320896)

[**9. Az oldalon elérhető funkciók:** 15](#_Toc131320897)

[Bejelentkezés és regisztráció: 15](#_Toc131320898)

[Főoldal: 16](#_Toc131320899)

[Burgereink: 17](#_Toc131320900)

[Kosár: 18](#_Toc131320901)

[**10. Admin felület: 20**](#_Toc131320902)

[Admin felület kinézete: 20](#_Toc131320903)

[Új hamburger feltöltése: 21](#_Toc131320904)

[**11. Szakdolgozat célja:** 22](#_Toc131320905)

[Megvalósítás és megállapítások: 22](#_Toc131320906)

[Célkitűzések, amik nem sikerültek teljességgel 22](#_Toc131320907)

# 1. Bevezetés:

## Témaválasztás indoklása:

Azért választottam ezt a témát mert, bár az asztali alkalmazások közelebb állnak hozzám sokkal több gyakorlatom van weboldalakkal és úgy tudom hogy, pályakezdőként általában weboldalakkal kapcsolatos munkákat könnyebb találni így nem árthat egy kmolyabb webes projekt.

Sok gondolkodás, fejtörés után egy autókölcsönző elkészítését tűztem ki magamnak, viszont próbáltam nagyjából “sablonosan” felépíteni, így lehetséges hogy hasznát veszem a jövőben más projektekhez is. Miután jobban belevetettem magamat a weboldal létrehozásába hirtelen túl sok dolgot próbáltam egyszerre beletenni, ezért van egy-két félkész funkció a weboldalamon, de egészen büszke vagyok a végeredményre. A szakdolgozatom átalakítása nagyon könnyen kivitelezhető, néhány óra alatt lehetséges bármilyen más kölcsönzős témára, de akár szimpla webshopra is átalakítani. A weboldal külseje elég nyers, sajnos ilyen téren nem vagyok túl kreatív, de hála ennek, kellő képpen letisztult és felhasználóbarát is.

## Elméleti és gyakorlati jelentőség:

Alapvetően az oldal elkészítése annyi jelentősséggel bír, hogy megtanulhattam elkészíteni egy bejelentkezéses és regisztrációs felületet amely megmutatja miként működik a bejelentkezés és az ehhez tartozó munkamenet. A weboldal több Bootstrapes funkciót is tartalmaz például Bootstrap kártyákat, modális ablakot és menüt is. Maga a weboldal szerkezete a HTML leíró nyelvvel készült. A háttérben működő adatbázis, a regisztrációk és az autók adatainak tárolásához pedig MySql-t használtam.

# 2. Fejlesztői dokumentáció:

Egy weboldal elkészítéséhez nélkülözhetetlen volt az adatbázis kezelés amihez a felsorolt programokat használtam:

* A **phpMyAdmin Database Manager 5.2.0**-es verziója, ami az adatbázis létrehozásához és kezeléséhez szükséges.
* A **MySql Database 10.4.27**-es verziója, ami az oldalak hátterében lévő adatbázisát szolgáltatja.

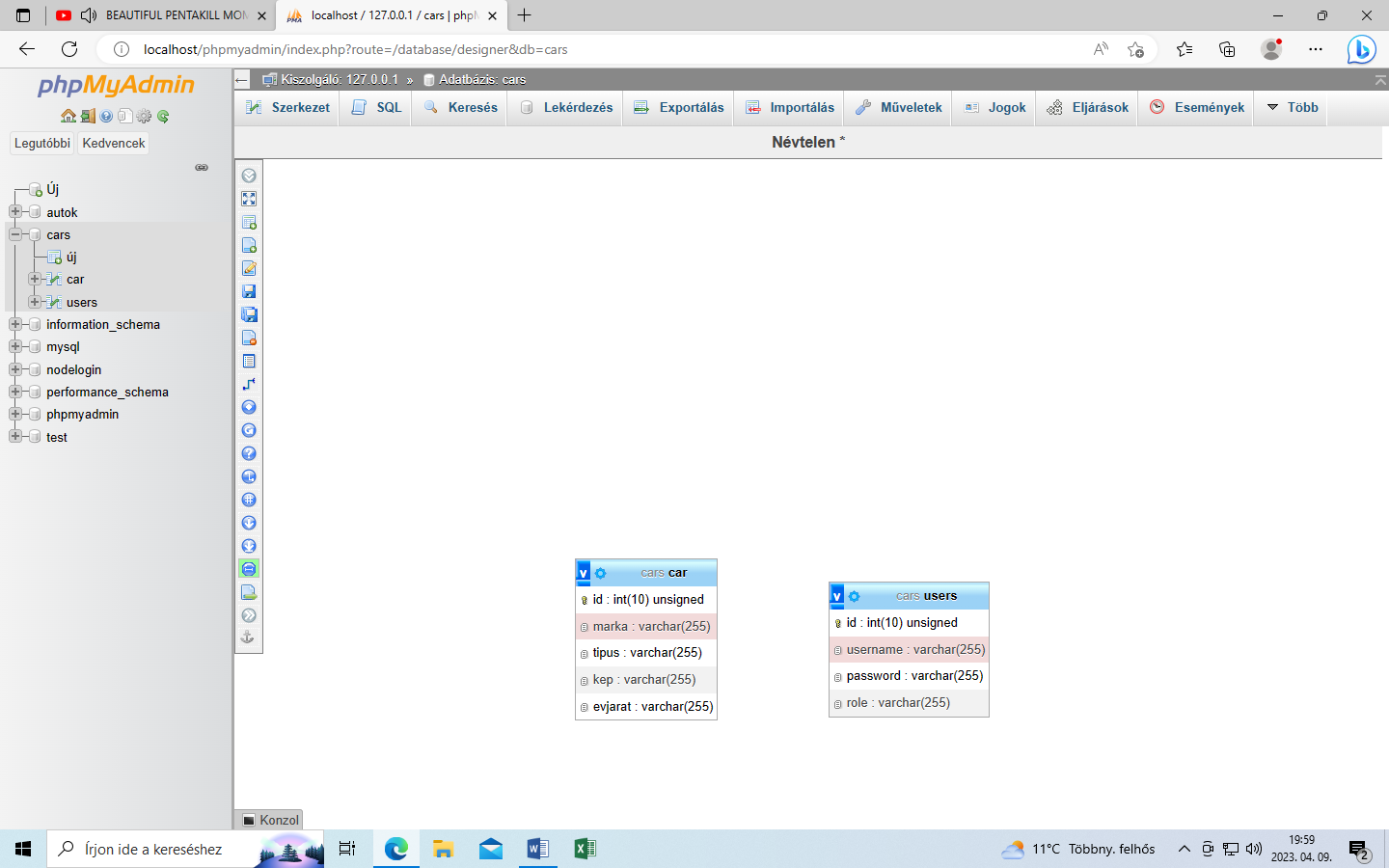
**Továbbá a weblap készítése során felhasználtam:**

* **Bootstrap 5** elemeket
* **NodeJs** konzol parancsokat
* **CSS** stílus lapot
* **JavaScript** kódokat

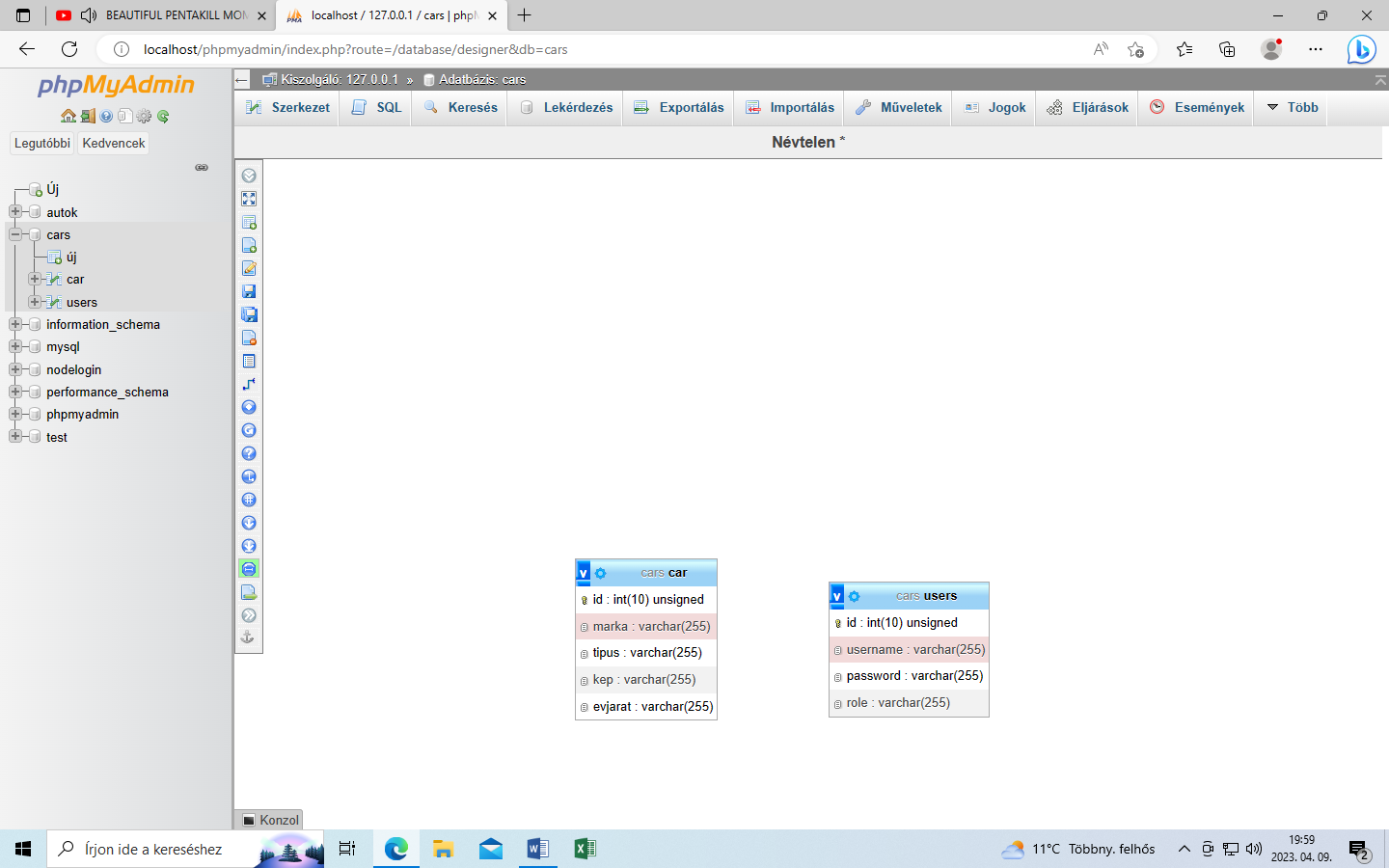
## Weblapon használt adatszerkezetek:

A weboldalam nem tárol túl sok adatot ezért nem éreztem szükségét sok különböző adattábla létrehozásának, lényegében az adatbázis két különböző, kapcsolat nélküli táblából áll, minden adat amelyet szükséges volt adatbázisban tárolni sima karakterláncként(varchar) került tárolásra, kivéve az id-k amelyek szám(INT, Integer) típusúak, ezek lényegében minden sornak egy egyedi sorszámot adnak, így könnyen meg lehet különböztetni két akár teljesen ugyan azt tartalmazó sort.

## Az adatbázis egyedi táblái és azok mezői:



## A users tábla:



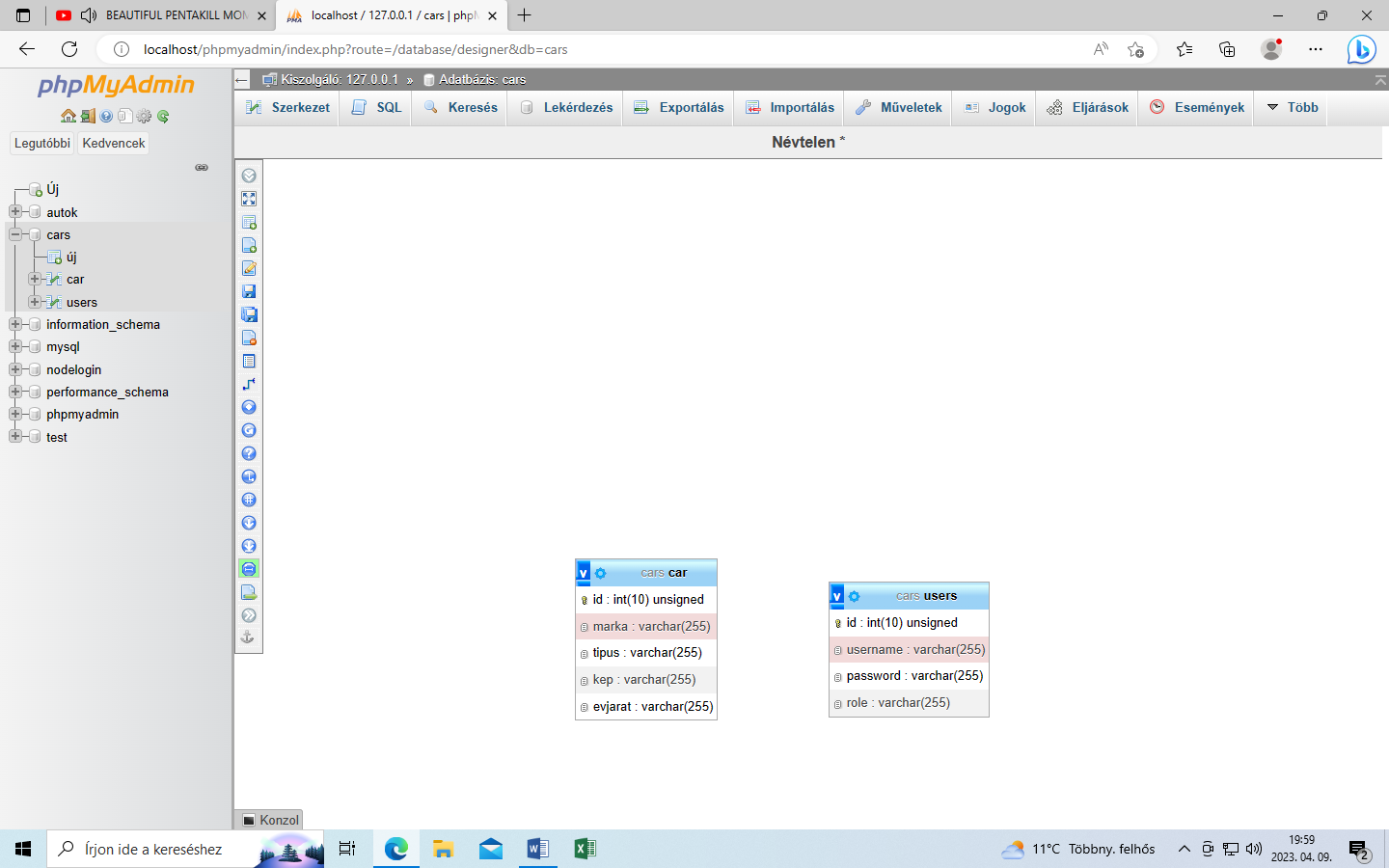
Elsődlegesen a felhasználók alap adatainak a tárolására szolgál. Első mezője az id szám(Integer) típusú, amely önmagától generálódik valahányszor egy újabb felhasználó regisztrál az oldalon és ha törlünk egy fiókot akkor hozzáigazítja az id-kat az új sorrendhez. Erre a tulajdonságra, oszlopra azért van szükség mert a későbbiek folyamán szükségünk lesz rá a felhasználók beléptetésekor az oldalon ezenkívül ez lesz a felhasználó azonosítója is és egyben ez különbözteti meg a aorokat(rekordokat) egymástól. A tábla második mezője a username a felhasználók neveinek tárolására szolgál szöveg típusban(varchar).  
A tábla harmadik mezője a password a felhasználók jelszavainak tárolását segíti elő mely titkosított formátumban jelenik meg az adatbázisban.

  
A tábla negyedik mezője a felhasználók rangját jelzi szöveg típusban˙(varchar), mely rangok a következőek lehetnek:

* user(Átlagos felhasználó)
* admin(Adminisztrátor)  
  

Lényegében ez a tábla az alapja a felhasználók regisztrációjának és magának a bejelentkezésnek.

## A cars tábla:



A tábla első mezője szintén egy integer típusú mező amely automatikusan egyedi azonosítót fűz a sorhoz, mivel automatikusan egyedi értéket ad ezért, ha új rekord kerül a táblába akkor az is egyedi azonosítóval lesz ellátva.

A tábla második mezője nem más mint maga a márka neve, amely egy varchar típusú mezőben lett eltárolva.

A tábla harmadik mezője az autó típusát tartalmazza, amely egy integer típusú mezőben lett eltárolva.

A tábla negyedik mezője az autóhoz feltöltött kép eléréséhez szükséges, a kép nevét tartalmazza egy varchar típusú mezőben.

A tábla utolsó mezője az autó évjáratát tartalmazza, ezt is varchar típussal tároltattam el az autóval mivel nem volt szükség integerre az oldal jelenlegi funkcióihoz és így valamivel egyszerűbb tárolni.

# 3 .Egyedi algoritmusok a weboldalon:

A következő algoritmusok a program megírásának sorrendjében vannak bemutatva itt a dokumentációban.

## A regisztráció:

Célja, hogy a felhasználónak legyen lehetősége regisztrálni, majd ezt felhasználva a beléptetés algoritmusával tovább kerüljön a főoldalra ahol ténylegesen tudja kezelni az oldalt. A lépések sorrendben:

* A felhasználó által kötelezően kitöltendő mezők (felhasználónév, jelszó) ellenőrzése, nem maradt-e üres szövegdoboz, ha igen hibaüzenet küldése.
* A használni kívánt felhasználónév leellenőrzése egy eldöntés tétellel hogy, van-e már használatban az oldalon ugyan az a felhasználónév, ha már regisztrált be valaki akkor hibaüzenet küldése.
* Hogyha a felhasználó mind a két kötelező mezőt kitöltötte, akkor a jelszó titkosítva lesz tárolva az adatbázisba amihez **bcryptet** használtam.

## A bejelentkezés algoritmusa:

A beléptetés az adatbázisból lekéri a bejelentkezni kívánó személy felhasználó nevét és jelszavát, amennyiben valamelyik nem egyezik akkor egy hiba üzenetet kapunk viszont, ha mind a kettő megfelelő akkor az oldal belépteti és azonnal tovább viszi a főoldalra.  
**A beléptetés kódja:**

app.post('/login', (req, res) => {

    const { username, password } = req.body;

    connection.query('SELECT \* FROM users WHERE username = ?', [username], (error, results) => {

        if (error) throw error;

        if (results.length === 0) {

            return res.status(400).json({ message: 'A felhasználónév vagy jelszó hibás!' });

        }

        const user = results[0];

        bcrypt.compare(password, user.password, (error, result) => {

            if (error) throw error;

            if (!result) {

                return res.status(400).json({ message: 'A felhasználónév vagy jelszó hibás!' });

            }

            req.session.username = username;

            req.session.role = user.role;

            res.json({ username: req.session.username, role: req.session.role });

        });

    });

});

## A kijelentkezés algoritmusa:

A kijelentkezésnél a weboldal megvizsgálja, hogy van-e bejelentkezett felhasználó, ha nincs hiba üzenetet kapunk viszont ha van akkor a felhasználó sikeresen kijelentkezhet.

A kijelentkezés kódja:

app.post('/logout', (req, res) => {

    if (!req.session.username) {

        return res.status(400).json({ message: 'Nincs bejelentkezett felhasználó' });

    }

    req.session.destroy(() => {

        res.json({ message: 'Sikeres kijelentkezés!' });

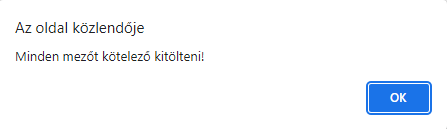
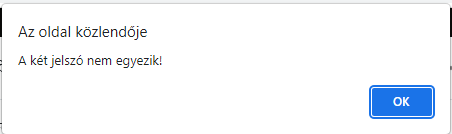
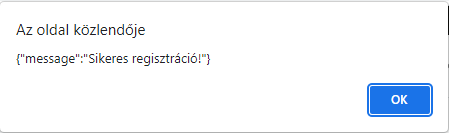
    });

});

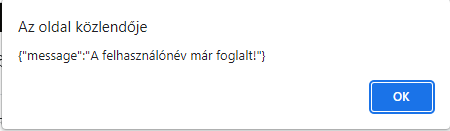
# 4. JavaScript funkciók:

## Regisztráció funkció a regisztráció gombhoz:

A regisztráció gombra kattintáskor megjelenik egy ablak melyben a program be kéri a felhasználónevet a jelszót, a jelszót ismét majd a program megvizsgálja, hogy a két beírt jelszó egyezik-e, illetve, hogy minden mező ki van-e töltve. Amennyiben a mezők nincsenek ki töltve vagy a két jelszó nem egyezik, abban az esetben ezek a hiba üzenetek fognak minket fogadni.

  
  
Viszont ha mindkét mezőt ki töltöttük és a két jelszavunk is megegyezik akkor már jöhet is a következő rendszer üzenet.  


Amennyiben viszont már felhasználónevünk foglalt ezt a kis hiba üzenetet kapjuk.



A regisztrációs felület a lehető összes problémára fel van készítve így útmutatást adhat a felhasználónak,hogy miért nem tud regisztrálni.  
**A regisztráció funkció kódja:**

function registration() {

    if (document.getElementById('regUsername').value != "" && document.getElementById('regPassword1').value != "" && document.getElementById('regPassword2').value != "") {

        if (document.getElementById('regPassword1').value == document.getElementById('regPassword2').value) {

            let datas = {

                username: document.getElementById("regUsername").value,

                password: document.getElementById("regPassword1").value,

            };

            let url = "http://localhost:3000/register";

            let fetchOptions = {

                method: "POST",

                body: JSON.stringify(datas),

                headers: { "Content-type": "application/json; charset=UTF-8" }

            };

            fetch(url, fetchOptions)

                .then(x => x.text())

                .then(y => {

                    alert(y);

                    document.getElementById('regUsername').value = "";

                    document.getElementById('regPassword1').value = "";

                    document.getElementById('regPassword2').value = "";

                });

        }

        else {

            alert("A két jelszó nem egyezik!");

        }

    }

    else {

        alert("Minden mezőt kötelező kitölteni!");

    }

}

## Felhasználó és rang kiírása:

Véleményem szerint nagyon fontos, hogy egy weboldalon ki legyen írva a felhasználó nevünk főleg, ha mindezt egy jó dizájn mellé tesszük. A program kódom bejelentkezésnél lekéri a felhasználó adatait majd ki iratja a weboldalra a felhasználónevét és rangját.

A 4. sor azért van a hogy akárhányszor betöltődik, újra töltődik az oldal ürítse ki a kereső mezőt mivel a mező magától feltöltődik a bejelentkezéses vagy regisztrációs adatokkal ha be van állítva az hogy a böngésző elmentse és magától beírja azokat.

**A felhasználó és a rang kiíratásának kódja:**

function load() {

    let username = sessionStorage.getItem("username");

    let role = sessionStorage.getItem("role");

    document.getElementById('searchingField').innerHTML="";

    if (role != "admin") {

        window.location.replace("index.html");

    }

    else {

        if (username) {

            let loggedInUser = document.getElementById("loggedInUser");

            if (loggedInUser) {

                loggedInUser.innerHTML = `${username} (${role})`;

            }

        }

    }

};

## Navigációs fül:

A navigációs fül az oldalak közötti navigációt segíti elő ez által jöhet létre például a fő oldalról a kölcsönzés oldalra navigálni. Ez a folyamat egy nagyon egyszerű kis JavaScript kód amit bárki könnyen tud alkalmazni.

function carRental() {

    window.location.replace('kolcsonzes.html')

}

Két oldal közti navigációhoz egy ilyen rövid function függvény szükséges.

## A keresés funkció és a kártyák megjelenítése az adatbázisból lekért adatokkal:

Ez egy elég összetett kódrészlet, de a lényege az hogy az első functionnel lekérjük a már kereső által szűrt adatokat és ezeket megjelenítjük a második function segítségével. Így hozható működésbe egyszerre a két funkció.

// Adatok lekérése a card-okhoz + autók közti keresés (szűkítés a megadott feltételeknek)

function cardDatas() {

    const columnNames = {

        1: 'marka',

        2: 'tipus',

        3: 'evjarat',

        4: 'kolcsonozve'

    }

    const searchingField = document.getElementById('searchingField').value;

    const searchType = columnNames[document.getElementById('searchType').value];

    console.log(searchingField);

    console.log(searchType);

    let datas = {

        searchingField: searchingField,

        searchType: searchType,

    };

    console.log(datas);

    let url = "http://localhost:3000/cards";

    let fetchOptions = {

        method: "POST",

        headers: {

            "Content-Type": "application/json"

        },

        body: JSON.stringify(datas)

    }

    console.log(fetchOptions);

    console.log(fetchOptions.headers);

    fetch(url, fetchOptions)

        .then(x => {

            console.log(x);

            return x.json();

        })

        .then(y => {

            console.log(y);

            cards(y);

        })

        .catch(error => console.error(error));

}

// Adatok kirajzoltatása

function cards(adatok) {

    let row = "";

    console.log(adatok);

    for (let adat of adatok) {

        row += `

        <div class="col-xxl-2 col-xl-3 col-md-4 col-sm-6 my-2">

            <div class="card h-100">

            <img src="http://localhost:3000/${adat.kep}" class="card-img-top img-fluid img-thumbnail" alt="${adat.kep}">

                <div class="card-body">

                    <h6 id=${adat.id} class="d-none">${adat.id}</h6>

                    <h5 class="card-title">${adat.marka}</h5>

                    <p class="card-text">${adat.tipus}</p>

                    <p class="card-text">${adat.evjarat}</p>

                    <p class="card-text">${adat.kolcsonozve}</p>

                    <button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#editModal" onclick="editableId = ${adat.id}">Módosítás</button>

                    <button type="button" class="btn btn-danger" onclick="deleteCar()">Törlés</button>

                </div>

            </div>

        </div>

        `;

    }

    document.getElementById('carCard').innerHTML = row;

}

# 5. Teszt dokumentáció:

## Az oldal tesztelése és a felmerülő hibák:

Az oldal le lett tesztelve 2 fajta böngészővel:

* Google Chrome
* Microsoft Edge

Változást a teszt alatt nem tapasztaltam minden teljesen ugyan olyan volt a két böngészőben a funkciók egyessével tesztelve lettek és minden egyformán működik a két böngészőben. Az oldal tesztelése során két ember segítségét kértem, egy osztálytársamat és a testvéremet aki nem tanult semmiféle programozást.

A két teszt során előkerült néhány hiba amelyek összegezve:

* A „kölcsönzés” gombra való kattintás nem csinált semmit, volt egy szemantikai(logikai) hiba a kódban ami miatt nem futott le az oldalak közötti ugráshoz szükséges function függvény.
* Egyik modális ablakban sem működött az „ok” és a „mégse” gomb, volt egy hiba amit mindenhol elkövettem, a két gomb nem volt valamiért része a modális ablaknak és a modális ablak működése miatt ami nem az ablak része az nem kattintható lényegében. Ezt egyszerűen meg lehetett oldani azzal hogy a két gombot a modális ablak div-jébe(keret) tettem bele.

<div class="modal fade" id="changePassword" data-bs-backdrop="static" data-bs-keyboard="false" tabindex="-1"

        aria-labelledby="staticBackdropLabel" aria-hidden="true">

        <div class="modal-dialog">

            <div class="modal-content">

                <div class="modal-header">

                    <h5 class="modal-title" id="staticBackdropLabel">Jelszó módosítása</h5>

                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

                </div>

                <div class="modal-body">

                    <label class="form-label">Új jelszó</label>

                    <input class="form-control" type="password" name="newPassword" id="newPassword" required>

                    <label class="form-label">Új jelszó megerősítése</label>

                    <input class="form-control" type="password" name="newPassword2" id="newPassword2" required>

                </div>

                <div class="modal-footer">

                    <button type="button" class="btn btn-primary" onclick="passwordChange()">Ok</button>

                    <button type="button" class="btn btn-danger" data-bs-dismiss="modal">Mégse</button>

                </div>

            </div>

        </div>

    </div>

# 6. Fejlesztési lehetőségek:

A programozásban egy programot szinte a végtelenségig lehet bővíteni, itt felsorolok egy pár fejlesztési lehetőséget a szakdolgozatommal kapcsolatban:

* Több elérhető nyelv
* A regisztráción belül születési idő, telefonszám, lakhely/értesítési cím megadása
* Az éppen kikölcsönzött autók elrejtése vagy jelölése a felhasználónak
* Admin felületen jelölni az éppen kikölcsönzött autókat és hogy kiknél vannak
* A weboldal külsejének formázása
* A kölcsönzés oldal bővítése

# 7. Hardware és szoftver igény:

Mivel ez egy webes felület ezért igényel egy eszközt, amivel lehet internetezni, ez lehet lap-top vagy asztali számítógép. Továbbá szükséges egy szerver, amire a weboldal kerül, ezen felül szükséges internet hozzáférés, valamint a következő böngészők egyike: Google Chrome, Microsoft Edge

## Telepítés és indítás:

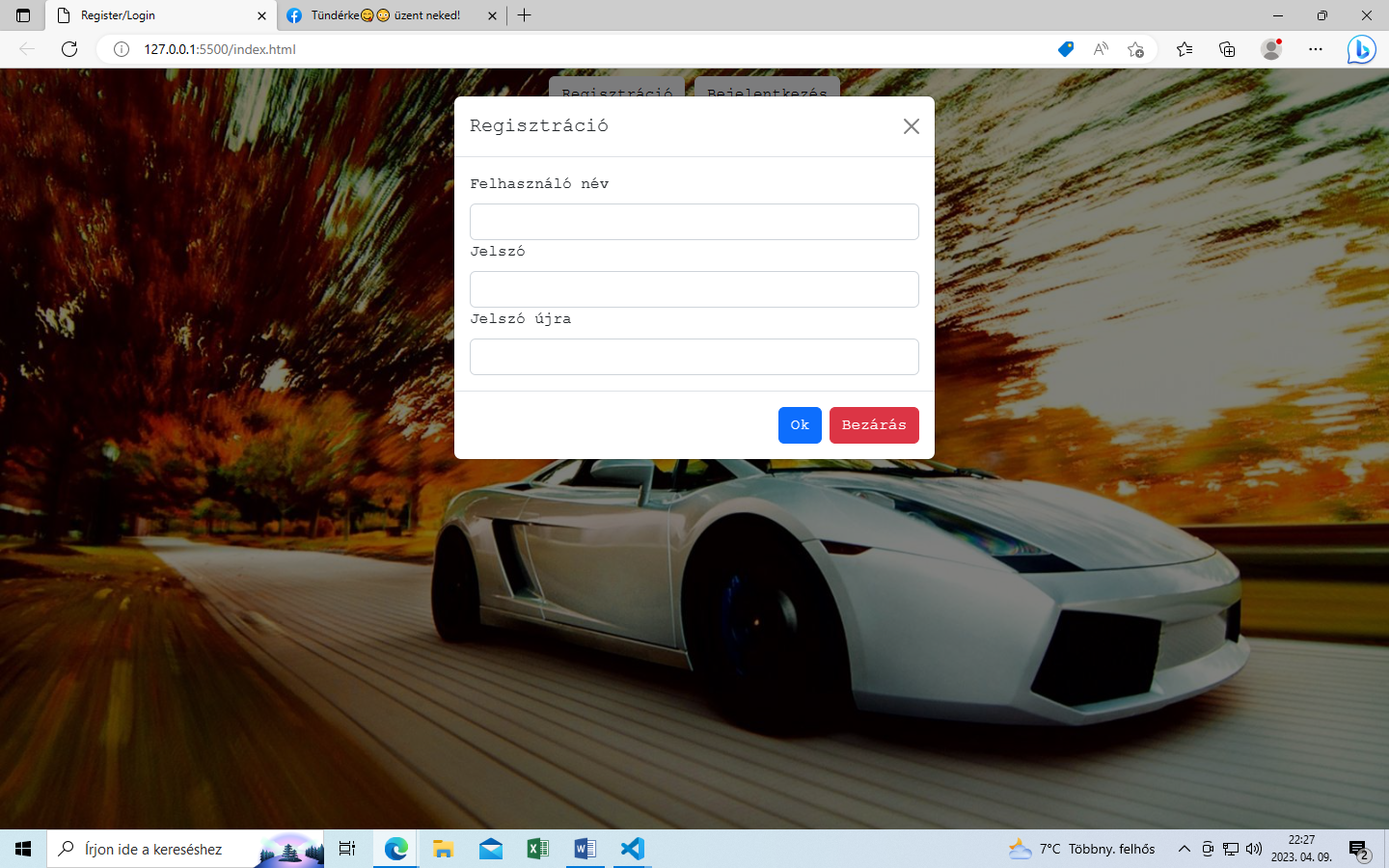
Mivel a szoftver webes alapú ezért külön telepítési eljárás csak abban az esetben szükséges, ha még nem rendelkezünk böngészővel. Szükséges még az oldal betöltéséhez internet hozzáférés is, amennyiben rendelkezünk ezzel a két dologgal már csak a megfelelő címet kell beírnunk

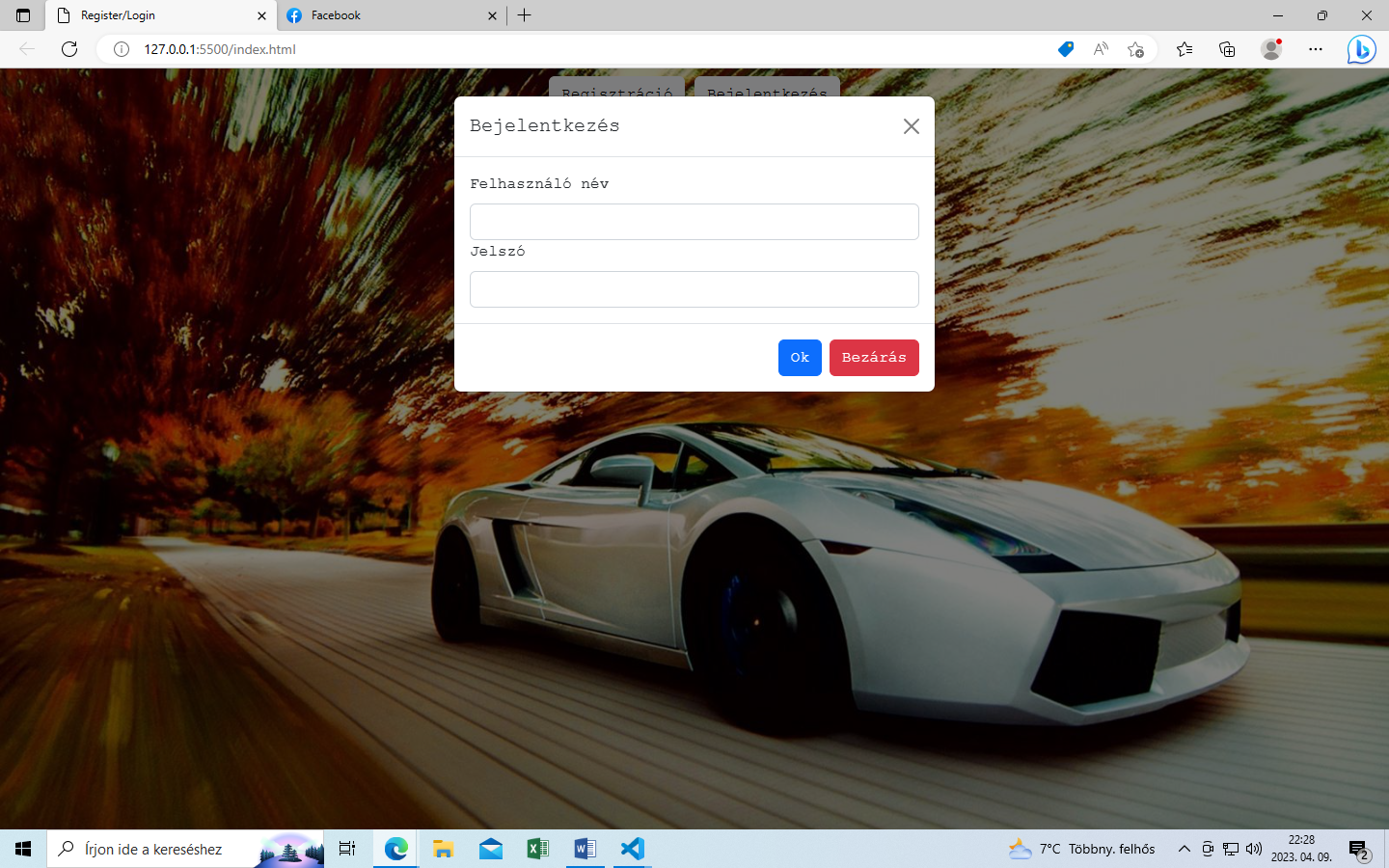
# 8. Az oldalon elérhető funkciók:

* A rólunk navigációs fülön való böngészés a helyi éttermek között
* Válogatás a különböző és lenyűgöző hamburgerek között
* Étel rendelés

## Bejelentkezés és regisztráció:

A regisztráció az oldalon való bejelentkezéshez és ez által az oldal funkcióinak használatához szükséges elengedhetetlen lépés. A funkció elérése a Bejelentkezési felületen keresztül történik, amely a weboldal jobb oldalán található, a Regisztráció gombon használatával. Erre kattintva a következő felületet kapjuk az oldal közepére tekintve.

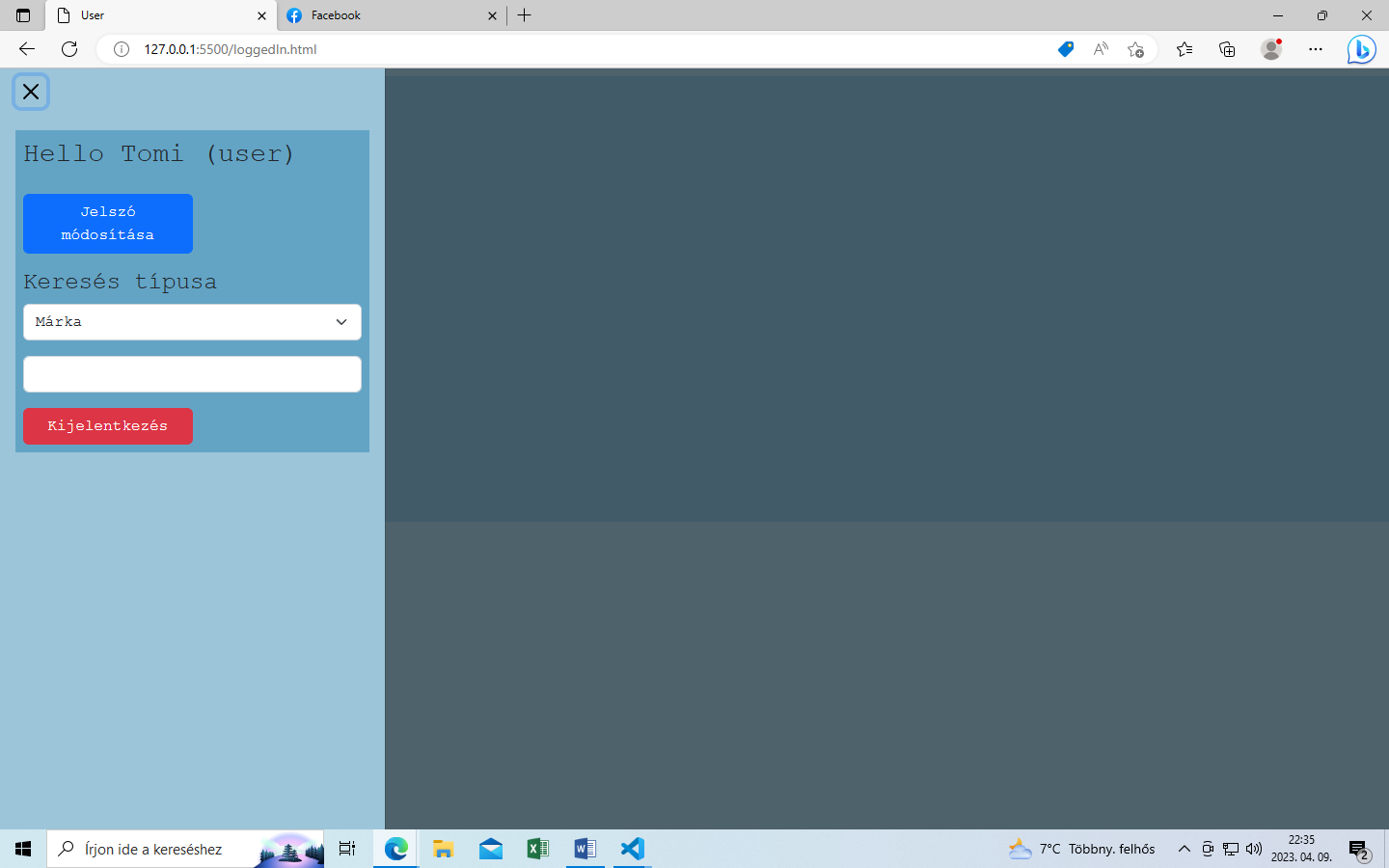




Mivel az eddigi ismereteim szerint az emberek jó része nem szeret a regisztrációval bajlódni vagy a kötelező regisztrációt meglátva elhagyja az oldalt, ezért egy egyszerűbb regisztrációt váronk csak el. Az egyszerű regisztráció érdekében csak a legszükségesebb dolgokat kérjük felhasználóinktól. Egy Felhasználó nevet, egy jelszót, illetve annak megerősítését.

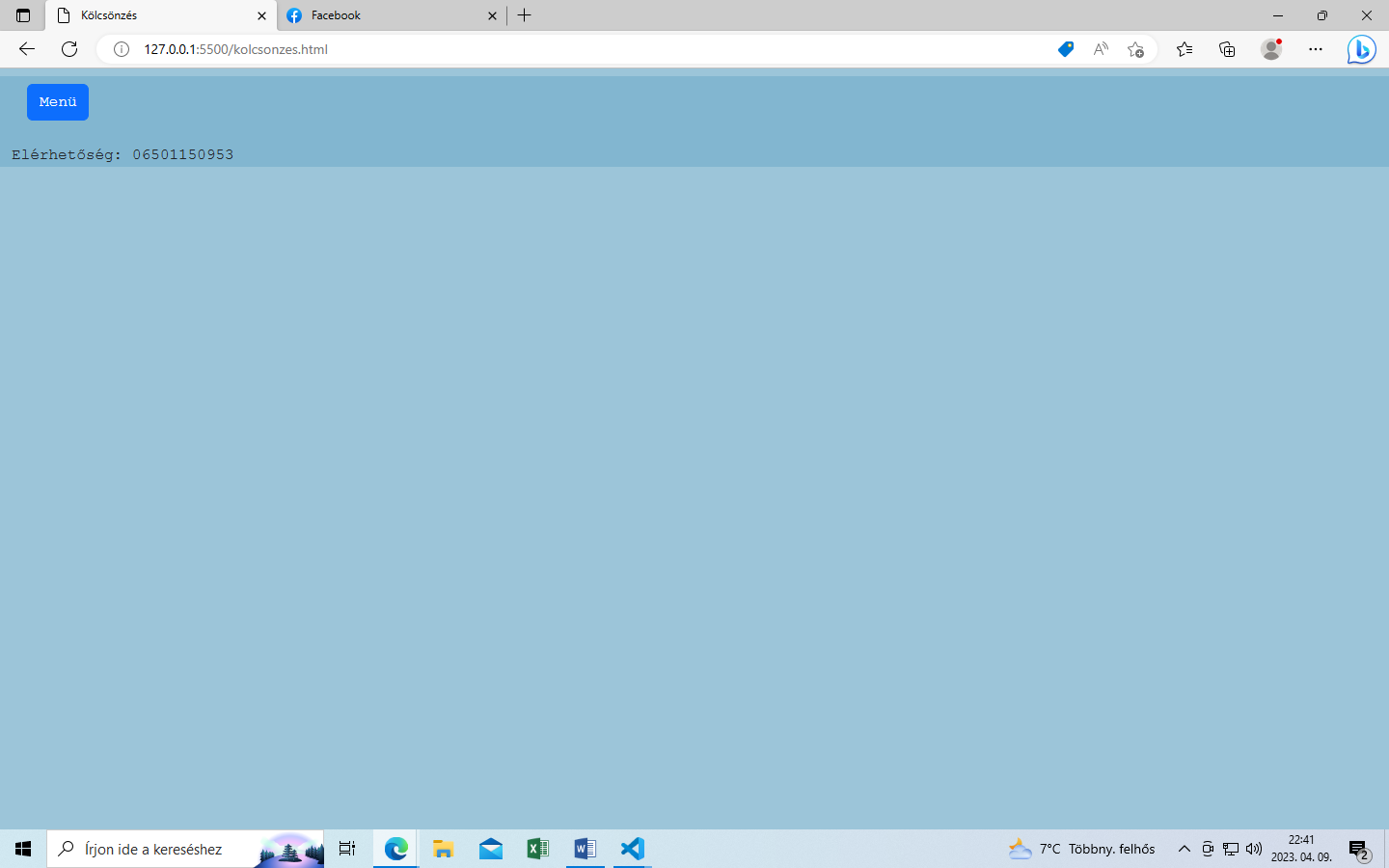
## Főoldal:

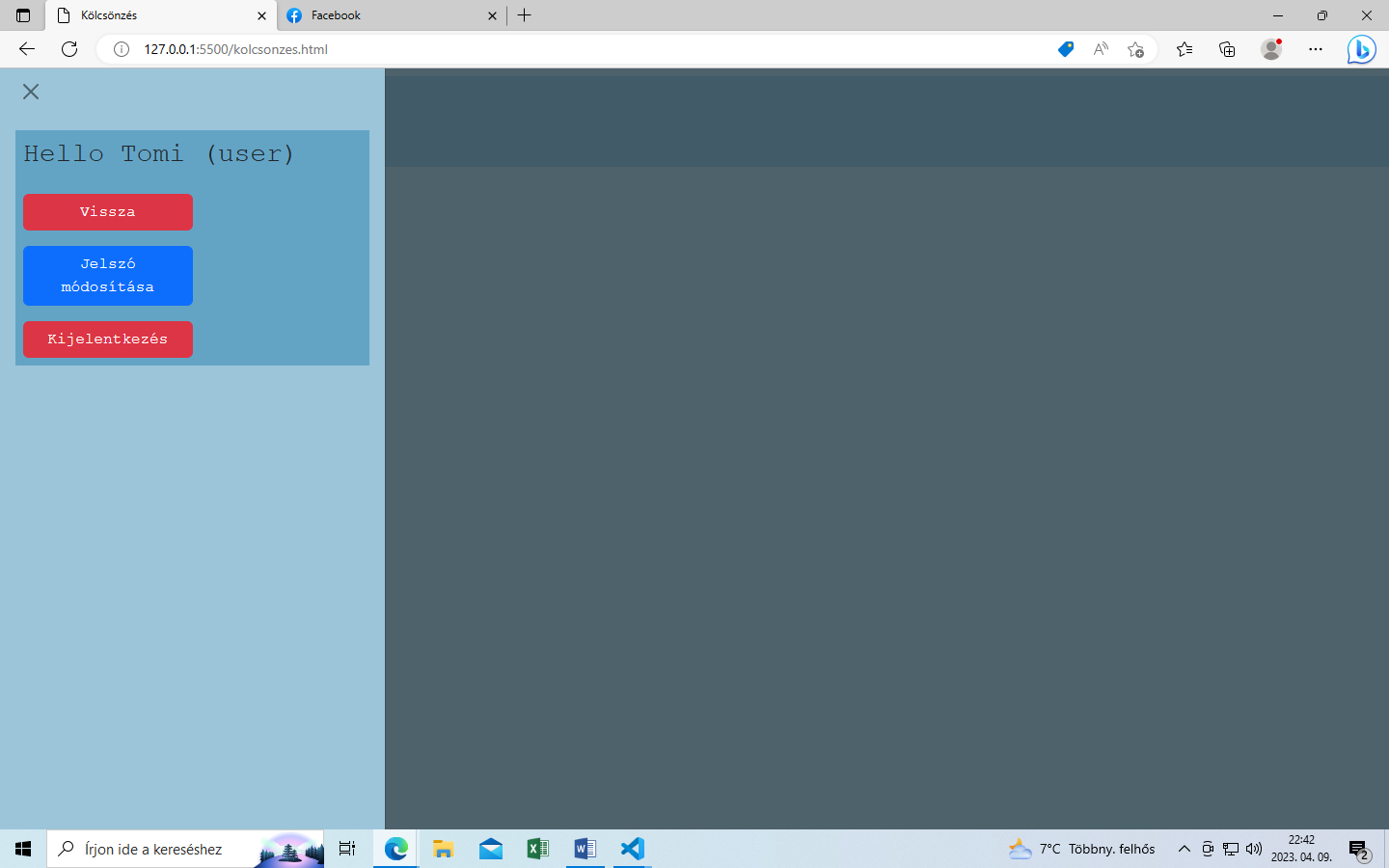
Bejelentkezés után a főoldal fogad minket ahol egyből láthatók is az elérhető autók és egy menü gomb. A menü gombra kattinva megjelenik egy függőleges sáv az oldal bal szélén ahol szűrést tudunk beállítani, ki tudunk jelentkezni, jelszót lehet módosítani és köszönt minket az oldal.



## A kölcsönzés oldal:

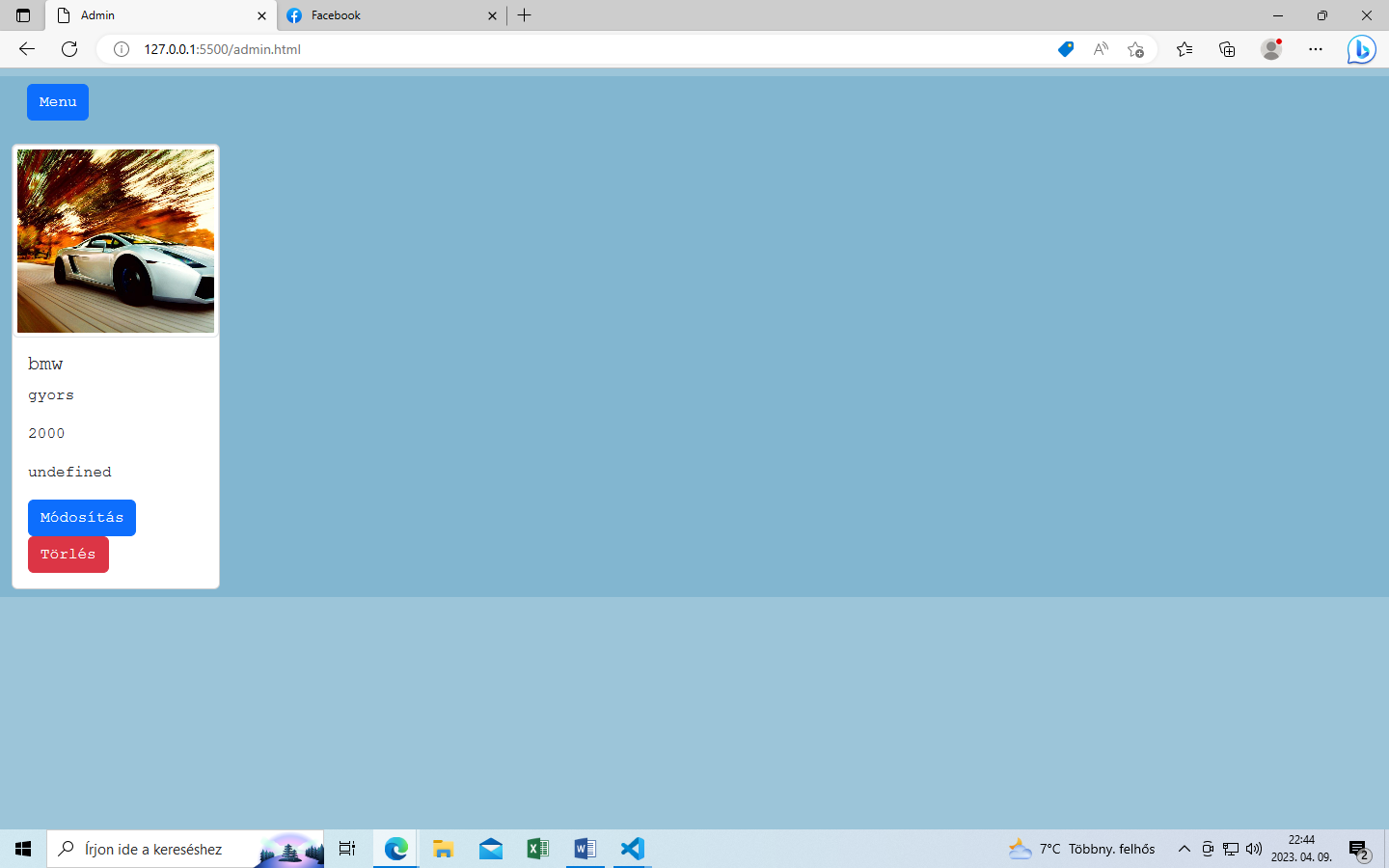
A kölcsönzés oldal egy szimpla elérhetőséget tartalmaz a kölcsönözni vágyók számára és egy menüpontot ahol vissza tudunk lépni a főoldalra, ki tudunk jelentkezni és jelszót tudunk módosítani.

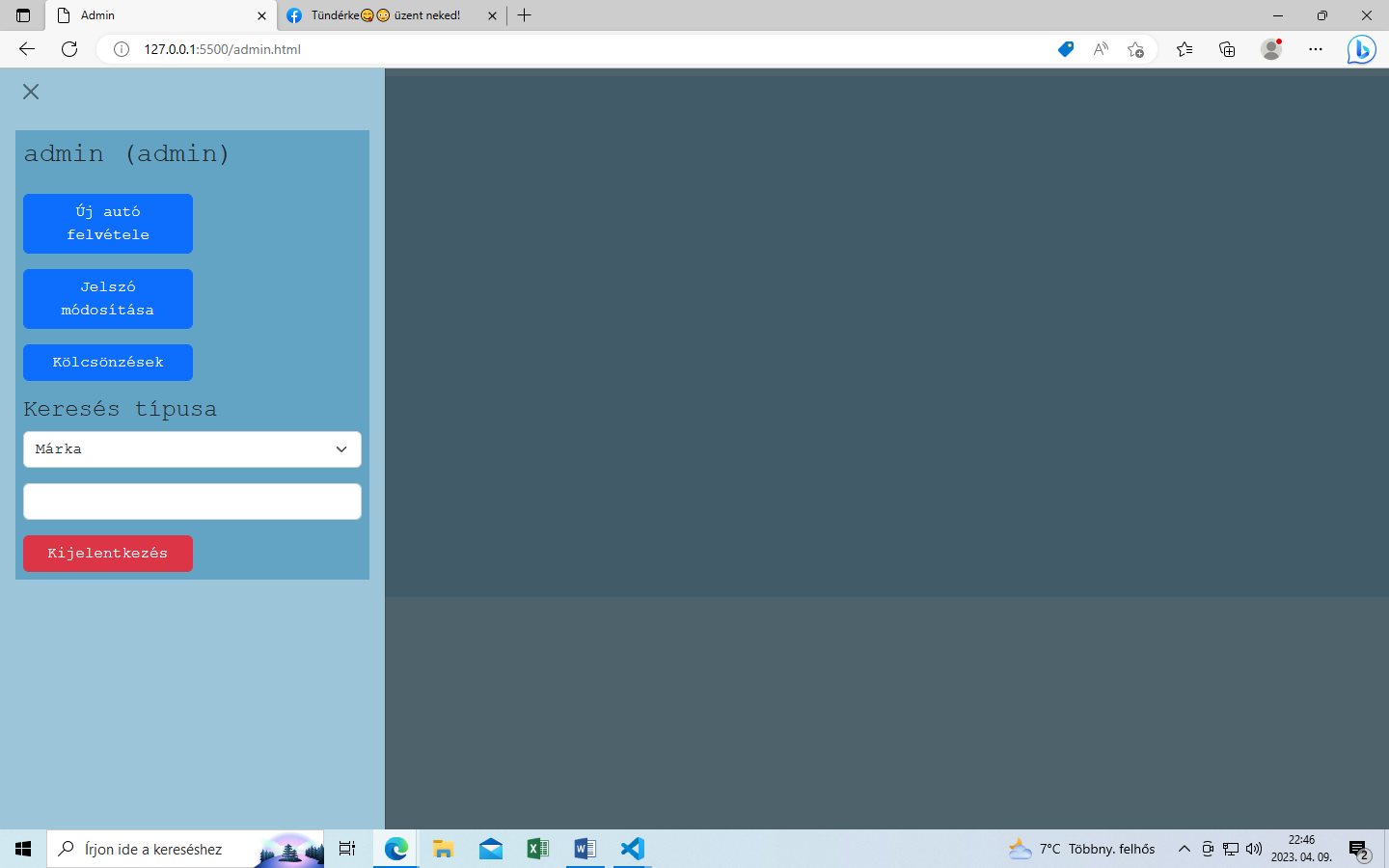


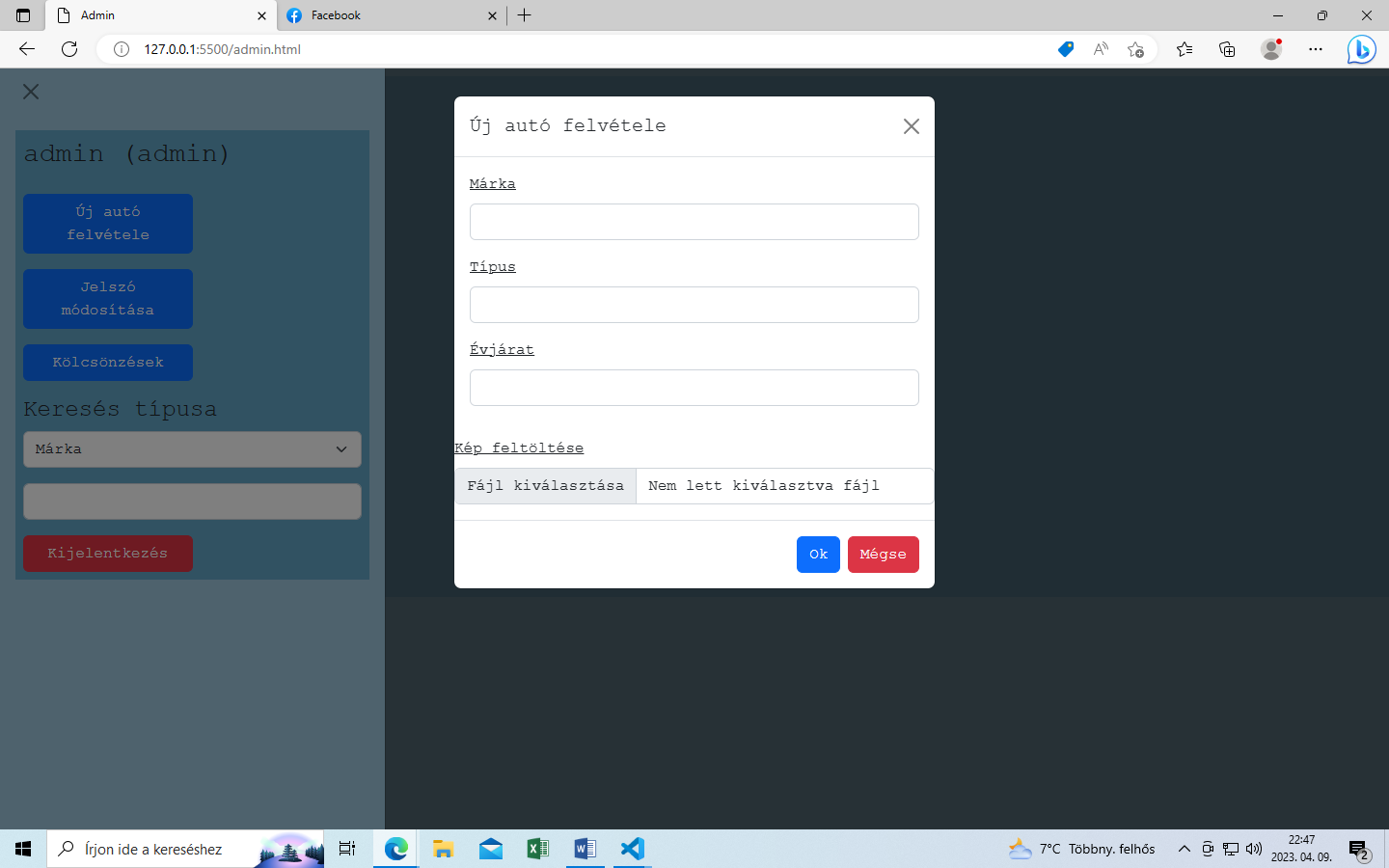


# 10. Admin felület:

## Az admin felület kinézete egyezik a felhasználó felület kinézetével viszont szerkesztési lehetőség van a kártyáknál és van lehetőség kártyát hozzáadni az oldalhoz de, akár kártyatörlésre is.







# 11. Szakdolgozat célja:

A szakdolgozatom célja egy autókölcsönző oldal volt több alapvető funkcióval ellátva hogy a jövőben lehetőség szerint tudjam sablonként használni.

## Megvalósítás és megállapítások:

Én azzal kezdtem az egészet hogy elképzeltem mi az amit szeretnék hogy tartalmazzon a weboldal és összeírtam a funkciókat. A következő lépés a weboldal szerkezetének felépítése volt, majd hozzáadtam egy minimális formázást a HTML kódon belül Bootstrappel és Css-sel. Ezután felépítettem az adatbázist majd Javascriptben létrehoztam a funkciókat hogy, életre kelhessen a weboldalam. Összességében elégedett vagyok azzal ami lett belőle, bár nem sikerült mindent megvalósítanom.

## Célkitűzések, amik nem sikerültek teljességgel

* A kölcsönzések oldalon szerettem volna megjeleníteni annak az autónak az adatait amelyik ki lett választva viszont ezt egyelőre nem sikerült kivitelezni.
* Szerettem volna még több funkciót hozzáadni az oldalhoz melyeknek a megvalósítása nem sikerült, például visszaigazoló e-mail, egy új oldal ahol korábbi kölcsönzők véleményt tudnak nyilvánítani, és segítséget kérni amely az admin felülettel lett volna kapcsolatban, lényegében egy chat ablak az összes user és az admin között.