



Vas Megyei Szakképzési Centrum
Nádasy Tamás Technikum és Kollégium

PROJEKTFELADAT

Körpörgő Autókereskedés

**Mészáros Bence, Bojsza Bulcsú, Jäger
Máté**

**Konzulens:
Balics Gábor**

2025

Nyilatkozat

Alulírott, Mészáros Bence, Bojsza Bulcsú, Jäger Máté kijelentem, hogy a Körpörgő Autókereskedés című projektfeladat kidolgozása a saját munkám, abban csak a megjelölt forrásokat, és a megjelölt mértékben használtam fel, az idézés szabályainak megfelelően, a hivatkozások pontos megjelölésével.

Eredményeim saját munkán, számításokon, kutatáson, valós méréseken alapulnak, és a legjobb tudásom szerint hitelek.

Csepreg, [beadás dátuma]

Hallgató

hallgató

hallgató

Kivonat

Körpörgő Autókereskedés

Jelen záradolgozat egy olyan webes alkalmazást mutat be, mely hatékony segítséget nyújt azon embereknek, akik új vagy használt autókat szeretnének találni, illetve azon embereknek, akik a saját tulajdonukban lévő autót szeretnék eladni.

Az oldal könnyű használatot nyújt mind a vevők, mint az eladók részére.

A vevők egyszerűen tudnak keresni és igényeiknek megfelelő autót találni. Ezen felül tudnak nekik tetsző autókat menteni a profiljukra.

Az eladók számára is számos előnyt nyújt az oldal mivel a feltöltés gyors és egyszerű, valamint a már feltöltött autók kezelése elérhető a saját profilon. Lehetőség van az oldalon fizetésre, hogy a hirdetett autó ki legyen emelve és elsőbbséget élvezzen a keresésnél.

Abstract

Körpörgő Car Marketplace

This final thesis presents a web application that provides an effective tool to help people who want to find new or used cars, and people who want to sell a car they own.

The site is easy to use for both buyers and sellers.

Buyers can easily search and find a car that suits their needs. They can also save cars they like to their profile.

For sellers, the site also offers a number of advantages as uploading is quick and easy, and the management of cars already uploaded is available on their profile. It is also possible to pay on the site so that the advertised car is highlighted and given priority in the search.

Tartalomjegyzék

1.	Bevezetés.....	8
1.1.	Témaválasztás indoklása.....	8
1.2.	Célkitűzés.....	8
1.3.	Kiknek szánjuk a weboldalt.....	8
1.4.	Hasonló weboldalak.....	8
1.5.	Fejlesztői környezet.....	9
2.	Felhasználói dokumentáció.....	10
2.1.	Hardware és software igény.....	10
2.1.1.	Minimális rendszerkövetelmények.....	10
2.1.2.	Ajánlott rendszerkövetelmények.....	10
2.1.3.	Ajánlott böngészők.....	10
2.2.	Főoldal.....	10
2.3.	Admin oldal.....	10
2.4.	Regisztráció.....	10
2.5.	Bejelentkezés.....	11
2.6.	Funkciók ismertetése.....	11
2.6.1.	Keresés.....	11
2.6.2.	Találatok.....	11
2.6.3.	Saját autó feltöltése eladásra.....	12
2.6.4.	Profil.....	12
2.6.5.	Fejléc.....	12
2.6.6.	Lábléc.....	12
3.	Fejlesztői dokumentáció.....	13
3.1.	Domain és tárhely.....	13

3.2.	Github és Git környezet	13
3.3.	HCaptcha	13
3.3.1.	A HCaptcha használatának haszna	13
3.3.2.	Működése	14
3.3.3.	Mire használjuk	14
3.4.	Prisma	14
3.4.1.	Miért használtuk	14
3.5.	BCrypt.....	14
3.6.	Stripe.....	15
3.6.1.	A fő szempontok a Stripe használata mellett.....	15
3.7.	Kialakított adatszerkezet és annak részletes bemutatása	15
3.7.1.	Adatbázis diagramm	15
3.7.2.	User Tábla.....	16
3.7.3.	Car_image_relation tábla.....	16
3.7.4.	Favorites tábla	16
3.7.5.	Car tábla.....	17
3.7.6.	Report tábla	18
3.7.7.	Összes kapcsolat az adatbázisban.....	18
3.8.	Components	18
3.8.1.	Header.....	19
3.8.2.	Footer.....	19
3.8.3.	CarsComponents.....	19
3.8.4.	ImageComponents	19
3.9.	Szerver-Kliens kapcsolat (API)	19
3.9.1.	API.....	19
3.10.	Jelszó titkosítás.....	20

3.10.1.	Hogyan működik.....	20
3.10.2.	Autentikáció.....	20
4.	Teszt dokumentáció	22
5.	A weblap design-ja	23
6.	Összefoglalás	24
6.1.	A szakdolgozat célja	24
6.2.	Megvalósítása	24
7.	Fejlesztési lehetőségek	26
7.1.	Több nyelv	26
7.2.	Több fajta jármű.....	26
7.3.	Adatok kezelése	26
8.	Irodalomjegyzék	27
9.	Mellékletek	28
9.1.	Components példák.....	28
9.1.1.	Keresés ár szerint.....	28
9.1.2.	Keresés meghaladás szerint.....	28
9.1.3.	Teljesítmény megadása feltöltéskor.....	28
9.1.4.	Kép feltöltés.....	28
9.1.5.	Ár megadása feltöltéskor	29
9.1.6.	Feltöltés limitáló	29

1. Bevezetés

Az autók és egyéb gépjárművek egyre nagyobb szerepet kapnak a mindennapokban és sokan el sem tudják képzelni az életüket nélkülük. Ma már nem ritka, hogy egy családnak több autó is van a birtokában. Szinte elengedhetetlen napjainkban, hogy autót használjunk. A legtöbb ember így megy munkába, iskolába vagy akár csak bevásárolni. Piacuk is egyre csak nő és nincs megállás. Folyamatosan új technológiákat találnak ki és fejlesztenek tovább. A használtautó piac egyre nagyobb érdeklődést vonz és évente közel egymillió használt autó cserél gazdát csak Magyarországon.

1.1. Témaválasztás indoklása

A weboldal ötlete egy hosszas gondolkodás eredménye. Az egyik csapattag vetette fel az ötletet mivel rajong az autókért és úgy érezte, hogy nincs hazánkban egy olyan online autókereskedés sem, ami megfelelő, könnyen kezelhető lenne mind a vásárlók mind az eladók számára.

1.2. Célkitűzés

A célunk egy átlátható és könnyen kezelhető autókereskedés megvalósítása, ami minden felhasználónak kényelmes megoldást nyújt. Szeretnénk ha mindenki a lehető legrövidebb időn belül találna meg a számára megfelelő gépjárművet. A weboldal lehetőséget nyújt keresésre, használt vagy akár teljesen új autó vásárlása iránt is, hiszen nem csak használt autót lehet vásárolni. Lehet saját profilt regisztrálni és így a saját eladóvá vált autót is lehet publikálni. A céljaink között volt a többnyelvűség is, hogy ne csak hazánkban tudják használni a felhasználók hanem világszerte.

1.3. Kiknek szánjuk a weboldalt

A weboldalt olyan embereknek szánjuk, akik máshol nem találnak megfelelő használt vagy új autót, illetve olyanoknak, akik el szeretnék adni a sajátjukat. A többnyelvűségnek köszönhetően itt nem csak bizonyos nyelven és országban lehet keresni és hirdetni, ami jelentősen növeli a sikeres autóvásárlás vagy autóeladás lehetőségét.

1.4. Hasonló weboldalak

A mi weboldalunkhoz legjobban a cars.com nevezetű külföldi oldal hasonlít, ami egy használt és új autót kínáló oldal. A hasznaltauto.hu mint ahogy a nevéből is adódik csak használt autók eladására szolgál. Csak magyar nyelven található és csak magyar hirdetések is találhatóak meg ami jelentősen lecsökkenti a választékot.

A mobile.de egy a hasznaltauto.hu-hoz hasonló oldal mivel csak használt járműveket árulnak. Ezeknél az oldalaknál az jelentheti a problémát, hogy nem nemzetközi és ezért egy szűkebb kínálatot találhat az ember.

1.5. Fejlesztői környezet

A weboldalunk elkészítéséhez különböző fejlesztőeszközöket használunk, amelyek segítenek a fejlesztésben és az adatok kezelésében. Elsőként említénénk meg a Visual Studio Code (VS Code) alkalmazást, amely sok pozitívummal és előnnyel rendelkezik. Könnyű használata, átláthatósága, valamint egyéb egyedi tulajdonsága nagyon könnyűvé tette a fejlesztést és nagyon felgyorsította a feladat elkészülési idejét. Másodsorban említénénk meg a Laragon-t, ami tartalmazza az Apache webkiszolgálót így lehetővé teszi a lokális szerver futtatását, illetve a MySQL adatbázis szerkesztését teszi lehetővé, valamint nagyon átlátható. Ezeken fölül tartalmaz egy saját terminált is, ami szintén a fejlesztés gyorsítását teszi lehetővé. Illetve tartalmazza a phpMyAdmin-t a MySQL webes irányítására. A MySQL adatbázis-kezelő rendszer tárolja az autókhoz, felhasználókhoz és egyéb szükséges adatokhoz tartozó információkat. Azért esett ezen alkalmazásokra a választásunk mivel ezekkel tanultunk és ismerjük őket. Nagyon megkönnyítette a fejlesztést, a tesztelést, valamint az adatbázis kezelését, hogy már általunk ismert alkalmazásokban fejleszthettünk. A weboldal dizájnjának kialakítására a TailWind CSS-t használtuk, ami egy nagyon könnyen és gyorsan használható stílusleíró nyelv mely rengetek innovatív lehetőséget nyújt a többi hozzá hasonló stílusleíró nyelvhez képest mint például a Bootstrap. A tesztelésre Postmant használtunk, ami lehetővé teszi az API-k gyors és könnyű tesztelését.

2. Felhasználói dokumentáció

2.1. Hardware és software igény

2.1.1. Minimális rendszerkövetelmények

- Legalább 1 GHz-es processzor
- 1 GB szabad memória
- Windows 7 vagy újabb operációs rendszer

2.1.2. Ajánlott rendszerkövetelmények

- Legalább 2 GHz-es processzor
- 2 GB szabad memória
- Windows 10 vagy újabb operációs rendszer

2.1.3. Ajánlott böngészők

- Google Chrome
- Mozilla Firefox
- Microsoft Edge
- Apple Safari
- Opera
- Brave
- Vivaldi

2.2. Főoldal

Minden fontos információ, ami egy látogató számára szükséges lehet megtalálható és minden oldal, amit szeretnének megtekinteni itt megtalálható vagy a fejlécben vagy a láblécben. A legnépszerűbb márkákat is el lehetett itt érni a keresés megnyitása nélkül.

2.3. Admin oldal

Az oldalon vannak admin fiókok melyeknek vannak a sima felhasználók jogosultságain kívül egyéb jogosultságuk. Ezek közé tartozik a felhasználók tiltása, hirdetések törlése, a jelentésekre válaszolás, valamint végtelen hirdetés feltöltési lehetőség. Ezen lehetőségeket egy külön admin oldalon érhetik el, amit egyszerű felhasználók nem láthatnak.

2.4. Regisztráció

Ahhoz, hogy regisztrálni tudjunk az oldalra a „Bejelentkezés/Regisztráció” fülre kell

kattintani a fejlécben. Itt ki kell tölteni minden adatot kötelezően. Miután minden adatot kitöltöttünk egy HCaptcha fog megjelenni, amit meg kell oldani és ki kell tölteni. Ha ezt is megoldottuk akkor már csak rá kell kattintani a regisztráció gombra és már létre is hoztuk a fiókot.

(--KÉP--)

A regisztráció kötelező amennyiben új autót akarunk feltenni az oldalra vagy pedig vevőként el akarunk menteni egy hirdetést.

2.5. Bejelentkezés

Ha már van regisztrált fiókunk akkor lehet bejelentkezni. A „Bejelentkezés/Regisztráció” gombra kattintva mely a fejlécben található lehetséges bejelentkezni. Itt nem kell megadni minden adatot, amit a regisztrációnál szükséges csak az E-mail címet és a jelszót.

Amennyiben nincs még fiókunk akkor a „Nincs még fiókod? Csinálj egyet!” felíratra kell kattintani és máris tudunk regisztrálni.

2.6. Funkciók ismertetése

2.6.1. Keresés

Ha autót akarunk nézni az első teendő, hogy a keresési szempontok szerint szűkítsük a találatokat. Ezt megtehetjük márka, modell, évjárat, motor típus, ár és egyéb adatok alapján. Ha rámentünk a keresésre akkor kilistázza az összes olyan járművet, amelyek egyeznek a keresési feltételekkel. Ha nincs találat nem jelenít meg semmit.

2.6.2. Találatok

Ha a keresés sikeres volt és találtunk olyan autót, amely tetszik, akkor rá tudunk kattintani és minden adatát meg tudjuk tekinteni, meg tudjuk nézni a csatolt képeket, valamint elolvashatjuk a leírását, amit a hirdető írt.

(--KÉP--)

A képeket meg lehet tekinteni nagyban és kicsiben is. A teljesítmény nincs külön kiírva viszont, ha rávisszük az egérmutatót a kilowatt-ra akkor kiírja a pontos lóerő értéket. Ezen a felületen viszont nem csak az autó, hanem az eladó adatait is meg tudjuk tekinteni. Ha a hirdetés megtetszik elmenthetjük a kedvenceik közé egy kattintással, ha viszont valami hibát vagy megfelelő hirdetést találunk esetleg nem megfelelő az autó leírása jelenthetjük a

hirdetést melyet egy admin bármikor felülbírállhat és amennyiben jogos volt a jelentés tilthatja a hirdetést vagy akár magát a hirdetőt is.

2.6.3. Saját autó feltöltése eladásra

Miután rákattintunk a „Feltöltés” gombra bejön a felület, ahol a mi eladásra kívánt autónkról tudjuk az összes információt feltölteni.

(--KÉP--)

Mindenekelőtt be kell jelentkezünk vagy ha még nincs profilunk létre kell hozni egyet. Ha már van profilunk akkor a képen látható minden adatot meg kell adni, illetve képeket kell hozzáadni. Lehet leírást is írni, de ez nem kötelező. A felszereltség részen nem kötelező hozzáadni semmit, amennyiben nem rendelkezik az adott sajátosságokkal. Ha minden adatot megadtunk és leírást is írtunk akkor már csak el kell küldeni és már el is lesz mentve. A feltett autókat bármikor meg lehet tekinteni a saját profilon és akár szerkeszteni vagy törölni is lehet a hirdetést.

2.6.4. Profil

Nagyon fontos szerepe van az oldalon a saját profil megtekintésének. Minden személyes adatot itt tudunk megtalálni. Itt láthatóak a feladott hirdetések vagy ha csak vevőként használjuk az oldalt akkor pedig a „Mentett hirdetések” gombra kattintva megtekinthetjük azon hirdetésekkel melyeket ez előtt elmentettünk. A profilon egyaránt lehet módosítani a saját adatokat valamint a feladott hirdetések adatait is. Ha már sikerült eladnunk az árult autót, vagy csak meggondoltuk magunkat akár törölni is itt tudjuk a hirdetésünket.

2.6.5. Fejléc

Található egy fejléc, ami tartalmazza a Bejelentkezés/Regisztráció, Főoldal, Kereskedés, Feltöltés, Autók menüpontokat. Mindegyik egy külön oldalra vezet. Ez minden oldalon meg van jelenítve.

2.6.6. Lábléc

Csakúgy, mint a fejléc ez is minden oldalon megtalálható a weboldal alján és a weboldal nevét, logóját, elérhetőségeket, közösségi oldalakat, valamint Copyright-ot. Ezen felül tartalmazza a Tailwind CSS, valamint a Flowbite eredeti url-jeit melyeket a készítés folyamán használtunk.

3. Fejlesztői dokumentáció

3.1. Domain és tárhely

A hostot-t az OVH szolgáltatótól, valamint a domain-t a rackhost-tól szereztük. A szervergép egy Intel Core processzorral 2 GB RAM-mal és 1 vCore-al rendelkezik és Frankfurtban helyezkedik el. Ezen felül a tárhely 20 GB. Mivel egy olcsó domaint kerestünk így a rackhost és szintén olcsó tárhelyet így az OVH szolgáltató tökéletes volt a számunkra és minden igényünket kielégítette.

3.2. Github és Git környezet

A kiírt igénynek eleget téve és közös munkánk zökkenőmentes végzése miatt, kialakítottunk egy GitHub környezetet a fejlesztésünknek. A környezetet az egyik csapattag egy saját repository-t hozott létre „szakdoga” néven és meghívta a többi csapattagot. Mindenki munkája könnyen követhetővé vált és mindenki látta a másik munkáját szinte azonnal és a fejlesztés során létrejött egyéb fájlok közös elérhetősége is megoldhatóvá vált. Az elkészült munka, tartalmazza az adatbázist, a programkódot és a dokumentációt, a következő GitHub repository-ban érhető el:

<https://github.com/Zeunig/szakdoga>. Itt megtalálható az összes szükséges anyag, a projekt teljes körű megértéséhez.

3.3. HCaptcha

A HCaptcha használata lehetővé teszi a botok és emberi csalók kiszűrését.

3.3.1. A HCaptcha használatának haszna

A CAPTCHA-teszt célja annak megállapítása, hogy az online felhasználó valóban ember-e, és nem bot. A CAPTCHA egy rövidítés, amely a „Completely Automated Public Turing test to tell Computers and Humans Apart” (Teljesen automatizált nyilvános Turing-teszt a számítógépek és az emberek megkülönböztetésére) rövidítése. A felhasználók gyakran találkoznak CAPTCHA és reCAPTCHA tesztekkel az interneten. Az ilyen tesztek a botok tevékenységének egyik módja, bár a megközelítésnek megvannak a maga hátrányai. Bár a CAPTCHA-kat az automatizált botok blokkolására tervezték, a CAPTCHA-k maguk is automatizáltak. Úgy vannak programozva, hogy egy weboldal bizonyos helyein

felbukkanjanak, és automatikusan átmennek vagy nem mennek át a felhasználókon.

3.3.2. Működése

A klasszikus CAPTCHA-k megoldásához a felhasználóknak betűket kell azonosítaniuk. A betűket úgy torzítják el, hogy a robotok nem valószínű, hogy képesek lesznek azonosítani őket. A teszt sikeres elvégzéséhez a felhasználóknak értelmezniük kell a torzított szöveget, be kell írniuk a helyes betűket egy űrlapmezőbe, és el kell küldeniük az űrlapot. Ha a betűk nem egyeznek, a felhasználóknak újra kell próbálkozniuk.

3.3.3. Mire használjuk

A felhasználóknak regisztrációkor meg kell oldaniuk egy HCaptcha-t amely megakadályozza a nem kívánt regisztrációkat és távol tartja a botokat valamint a káros felhasználókat.

3.4. Prisma

A „Prisma” egy újfajta Data Mapper ORM eszköz, amely interfészként szolgál a webes alkalmazás és az adatbázisa között. Az intuitív adatmodellnek, az automatikus migrációknak, a típusbiztonságnak és az automatikus kitöltésnek köszönhetően absztrahálja a szerverrel való kommunikáció bonyolultságát, és lehetővé teszi, hogy az alkalmazás építésére összpontosítson, ahelyett, hogy a következő SQL-lekérdezésen gondolkodna.

3.4.1. Miért használtuk

A gyors és egyszerű adatbázis használat érdekében használtuk, hogy ne keljen az adatbázis query-vel foglalkoznunk és tudjunk fókuszálni a webfejlesztés részre.

3.5. BCrypt

A Bcrypt egy méltán elismert jelszótároló megoldás, amely bonyolult kriptográfiai algoritmusokat használ, jelentősen csökkentve annak esélyét, hogy egy hacker feltörjön egy jelszót. A Bcrypt egy kriptográfiai hash-funkció, amelyet jelszavak hashelésére és biztonságos tárolására terveztek az alkalmazások háttértárában úgy, hogy az kevésbé legyen érzékeny a szótáralapú kibertámadásokra. A bcrypt ahelyett, hogy egyszerűen hash-eli az adott jelszót, egy véletlenszerű adatot, úgynevezett só-t ad hozzá, hogy olyan egyedi hash-t hozzon létre, amelyet szinte lehetetlen feltörni automatikus találgatásokkal a hash-szótár és a nyers erővel végrehajtott támadások során.

3.6. Stripe

Az eladók fizethetnek, hogy az általuk hirdetett autó a kiemeltek között szerepeljen, ami azt jelenti, hogy előrébb lesz a kilistázásnál a többivel szemben. Ennek a lehetőségnek a megvalósítására a Stripe-ot alkalmaztuk, ami segít lebonyolítani a fizetést a felhasználók és a weboldalhoz kötött bankszámla között. Ez egy nagyon fontos dolog a weboldal szempontjából mert e nélkül nagyon nehéz lenne vele pénz keresni.

3.6.1. A fő szempontok a Stripe használata mellett

- Jó online és offline fizetések lebonyolítására: A Stripe-ot széles körben használják online fizetések feldolgozására, de nem csak erre jó. Ha rendelkezik fizikai hellyel, akkor képes lesz fizetéseket fogadni egy terminálon keresztül ugyanazzal a szolgáltatással, amelyet online fizetéshez használ.
- Mindenütt jelen van: A Stripe a fizetésfeldolgozási piac több mint egyharmadát birtokolja, mindössze néhány százalékponttal lemaradva a PayPal mögött. Használatának megtanulása hasznos készség, függetlenül attól, hogy milyen projekten dolgozol.
- Egyszerű használat: Csupán néhány lépéssel megkezdheti a kifizetések fogadását a Stripe-tól, gyakran olyan egyszerű, mint egy hivatkozás hozzáadása a webhelyére. Ha fizetési feldolgozást kell hozzáadnia egy alkalmazáshoz vagy termékhez, az általában csak néhány sornyi kódot vesz igénybe.
- Alapból integrált: A Stripe rengeteg beépített integrációval rendelkezik a legtöbb szervezet által már használt eszközökkel, a Freshbookstól a HubSpotig és a Salesforce-ig. Így sokkal könnyebben beilleszthető a meglévő szerszámkészletbe.
- Több mind pénz átvívó: Míg a Stripe leginkább az ügyfelektől a vállalkozások felé mozgó platformként ismert, ennél sokkal többet tesz. Használható számlázásra, előfizetések beállítására, csalásmegelőzésre.

3.7. Kialakított adatszerkezet és annak részletes bemutatása

A weboldalunk adatbázisa 5 táblát tartalmaz. Ezek a táblák strukturáltak és szervezettek, hogy hatékonyan kezeljék és tárolják az adatokat. Ezek a táblák a „user”, „car”, „car_image_relation”, „report” és „favorites” néven találhatók meg.

3.7.1. Adatbázis diagramm

(--KÉP--)

3.7.2. User Tábla

A felhasználók adatainak tárolására szolgál

Mezőnév	Típus	Megjegyzés
id	INT, PK, NN, AI	Egyedi azonosító
name	VARCHAR, NN	A felhasználók neve
phone_number	VARCHAR	A felhasználók telefonszáma
email	VARCHAR, NN	A felhasználók e-mail címe
password	VARCHAR, NN	A felhasználók jelszava
avatar_url	VARCHAR	A felhasználók képeinek URL-je
join_date	TIMESTAMP	Csatlakozás időpontja
permissions	BIGINT, NN	A felhasználók jogosultságai
address	VARCHAR	A felhasználók címe

3.7.3. Car_image_relation tábla

Az autók képeinek tárolására szolgáló tábla

Mezőnév	Típus	Megjegyzés
id	INT, PK, NN, AI	Egyedi azonosító
car_id	INT	Az autó egyedi azonosítója
image_url	VARCHAR	Az autók képeinek URL-jei

3.7.4. Favorites tábla

A felhasználók saját elmentett autóhirdetéseinek tárolására szolgál

Mezőnév	Típus	Megjegyzés
id	INT, PK, NN, AI	Egyedi azonosító
car_id	INT	Az autók egyedi azonosítója
user_id	INT	A felhasználók egyedi azonosítója

3.7.5. Car tábla

Az autók tárolására szolgáló tábla

Mezőnév	Típus	Megjegyzés
id	INT, PK, NN, AI	Egyedi azonosító
creation_date	TIMESTAMP	A generálás dátuma
seller_id	INT, NN	Az eladó azonosítója
featured	TINYINT	Kiemelt-e az adott autó
price	INT, NN	Az autó ára
discounted_price	INT, NN	Az autó akciós ára
brand	VARCHAR, NN	Az autó márkája
model	VARCHAR, NN	Az autó modellje
year	INT	Az autó évjárata
mileage	INT, NN	Az autó Km/h állása
weight	INT, NN	Az autó súlya
horsepower	INT, NN	Az autó lóereje
cc	INT, NN	Az autó köbcentimétere
fuel_type	VARCHAR, NN	Az autó üzemanyag típusa
gearbox	VARCHAR, NN	Az autó váltójának típusa
drive_type	VARCHAR, NN	A meghajtott kerekek száma
condition	VARCHAR, NN	Az autó állapota
passengers	INT, NN	Ülések száma
doors	INT, NN	Ajtók száma
color	VARCHAR, NN	Az autó színe
features	BIGINT	Az autó egyéb sajátosságai
vin	VARCHAR, NN	Az autó alvázszáma
design	VARCHAR, NN	Az autó designja
listed	TINYINT, NN	Listázott-e az adott autó

description	TEXT	Az autók leírása
--------------------	------	------------------

3.7.6. Report tábla

A jelentett hirdetések adatait tárolja

Mezőnév	Típus	Megjegyzés
id	INT, NN	Az egyes jelentések egyedi azonosítója
car_id	INT, NN	A jelentett hirdetés azonosítója
reason	TEXT, NN	A jelentés indoklása
reporter_ip	VARCHAR, NN	A jelentő felhasználó
solved	TINYINT	Admin által felülbírált-e

3.7.7. Összes kapcsolat az adatbázisban

Car_image_relation tábla kapcsolatai

- car_image_relation.car_id -> car.id

Favorites tábla kapcsolatai

- favorites.car_id -> car.id
- favorites.user_id -> user.id

Car tábla kapcsolatai

- car.seller_id -> user.id

Report tábla kapcsolatai

- report.car_id -> car.id

3.8. Components

- Újra felhasználhatóság: Komponenseket lehet újra felhasználni különböző részekben és funkciókban a projektben, ami csökkenti a kód ismétlődését és segít az alkalmazás karbantarthatóságának növelésében.
- Modularitás: A komponensek moduláris szerkezetűek, ami lehetővé teszi, hogy az alkalmazást kisebb részekre bontsuk, így könnyebb megérteni és fejleszteni.

- Könnyű karbantarthatóság: Az egyes komponensek önállóan működnek, így könnyen lehet őket tesztelni, fejleszteni és javítani anélkül, hogy az egész alkalmazást érintené.
- Átláthatóság: A komponensek segítik a weblap strukturáltabbá tételét, mivel minden egyes részegység felelős egy adott funkcióért vagy feladatért.

A szakdolgozatunk készítése alatt sok komponenst használtam szám szerint (-----)db.

Név szerint a legfontosabbak

3.8.1. Header

- „Header”: Az oldal tetején elhelyezett navigációs sáv, amelyen keresztül a felhasználók könnyen elnavigálhatnak az oldal különböző részei között.

3.8.2. Footer

- „Footer”: Az oldal alján elhelyezkedő sáv, ahol a látogatók megtalálják az elérhetőségeinket

3.8.3. CarsComponents

- „CarCard”: Az egyes autók megjelenítésére szolgál
- „CarCardHL”: Az egyes autók megjelenítésére szolgál
- „RowCarCard”: Az egyes autók adatainak megjelenítésére szolgál
- „CarListing”: Az autók adatainak megjelenítésére szolgál miután rákattintottunk az egyes hirdetésekre
- „CarSearchCard”: Az autók keresési találatainak megjelenítésére szolgál
- „MyCars”: A felhasználók saját autóinak megtekintésére szolgál

3.8.4. ImageComponents

- „BigImageViewer”: A képek nagyban való megtekintését teszi lehetővé
- „ImageViewer”: A képek kicsiben való megtekintését teszi lehetővé

3.9. Szerver-Kliens kapcsolat (API)

3.9.1. API

Az API (Application Programming Interface) egy olyan interfész, amely lehetővé teszi az alkalmazások közötti kommunikációt. A szerver-kliens kapcsolat alapvetően az alábbiakon alapul:

- Protokoll: Általában HTTP vagy HTTPS protokollon keresztül történik a kommunikáció.
- Kérés-Felelet Modell: A kliens küld egy kérést a szervernek, ami válaszol a kérésre.

- Metódusok: „GET”, „POST”, „PUT”, „DELETE” stb. segítségével történik az adatok kezelése.
- Adatformátumok: Általában JSON vagy XML formátumban küldik az adatokat.
- Hitelesítés és Biztonság: Felhasználói hitelesítésre és adatbiztonságra szolgáló mechanizmusok alkalmazása.
- Dokumentáció: Részletes leírás a rendelkezésre álló végpontokról és használati utasításokról.
- Tesztelés és Hibakezelés: Az API-k alapos tesztelése és megfelelő hibakezelési mechanizmusok alkalmazása.

3.10. Jelszó titkosítás

A szakdolgozatunk készítése során a bcrypt-et használtuk a felhasználók jelszavaknak biztonságos tárolására és ellenőrzésére. Az alábbi kód részlete a backend része a server.js fájlban: A bcrypt egy kriptográfiai hash funkció, amelyet a jelszavak biztonságos tárolására és az azokkal való azonosításra használtunk. A funkció nagy előnye, hogy lassú, ami megnehezíti a brute force vagy rainbow támadásokat, ahol a támadók több próbálkozást tesznek a helyes jelszó megtalálására.

3.10.1. Hogyan működik

- Hash generálása: A jelszót több körben hasheli. Ezeket a köröket "költségszorzó" is nevezik. Minél magasabb a költségszorzó, annál hosszabb ideig tart a hashelés, és annál nehezebb a támadóknak feltörni a jelszót.
- Hash tárolása: A végső hash-távot tárolják, amely tartalmazza mind a sót, mind a hashelt jelszót. Ez az érték kerül mentésre az adatbázisba. A szakdolgozatomban használtam a felhasználók beléptetésére. A login API-ban a felhasználó által megadott jelszót a bcrypt segítségével hasonlítjuk össze a tárolt, hashelt jelszóval az adatbázisban. Ha a jelszavak megegyeznek, akkor JWT token-t generálok és visszaküldünk a felhasználónak, különben hibaüzenetet küldünk vissza.

3.10.2. Autentikáció

A szerver oldalon az autentikáció egy folyamat, amely a felhasználók beazonosítását és hitelesítését szolgálja. A folyamat a következő lépésekből áll:

- Felhasználó azonosítása: Amikor a felhasználó bejelentkezik, a szerver ellenőrzi az adatbázisban, hogy létezik-e a megadott felhasználónévvel rendelkező felhasználó.
- Jelszó ellenőrzése: Ha a felhasználó létezik az adatbázisban, a szerver ellenőrzi a megadott jelszót a tárolt jelszóval szemben. Ez általában egy hash-függvény használatával történik, hogy a jelszavak ne tárolódjanak nyílt szöveggént.

- Token létrehozása: Ha a jelszó helyes, a szerver létrehoz egy JSON Web Token (JWT) -t. Ebben a tokénben általában azonosító információk, például a felhasználó azonosítója (user_id), vannak kódolva. Válasz küldése a kliensnek: A szerver visszaküldi a kliensnek a JWT token. Ezt a token a kliens későbbi kérések során használja felhasználói azonosításra és hitelesítésre.
- Süti vagy más azonosító küldése: A szerver általában egy sütit küld vissza a kliensnek azonosítás céljából. Ez lehetőséget ad arra, hogy a kliens további kéréseket küldjön a szervernek anélkül, hogy újra be kellene jelentkeznie. Ezek az autentikáció fő lépései, amelyeket a server.js oldalon valósultja meg, míg a kliens oldalon ezeket a token kezeli és használja az azonosításhoz és a jogosultságok kezeléséhez.

Ezek az autentikáció fő lépései, amelyeket a server.js oldalon valósultja meg, míg a kliens oldalon ezeket a token kezeli és használja az azonosításhoz és a jogosultságok kezeléséhez

4. Tesztdokumentáció

A fejlesztés során rendszeresen teszteltük a kódot, és ezt párhuzamosan végeztük a fejlesztési folyamattal. Miután elkészült a program, többször is kipróbáltuk annak működését különböző helyzetekben, hogy meggyőződjünk a stabilitásáról és funkcionalitásáról. A keresést volt a legnehezebb tesztelni mivel itt volt a legtöbb lehetőség és legtöbb hibaforrás. A keresés mellett a feltöltés tesztelése jelentett kihívást. Az API-khoz történő kérések ellenőrzésére és validálására a terminálban consologok segítségével és postmanban tettük. Ezek a konzolüzenetek segítettek abban, hogy könnyedén futtathassuk és ellenőrizzük az összes API hívást, biztosítva azok helyes működését.

5. A weblap design-ja

A weblap kitalálásánál még nem tudtuk, hogy milyen kinézetűt szeretnénk csinálni. Az első 1 hónapot csak a design kitalálásával töltöttük és sokat gondolkozunk rajta. Mikor nekiálltunk a weblap elkészítésének már volt egy design és színpaletta, amit követtünk, de az eredeti gondolatoktól jelentősen eltérő lett a végkifejlet. Ahogy a weboldal készült vele együttemben változtak a designelemek.

A felhasznált ikonokat képeket flaticon-ról valamint freepik-ről és AI segítségével generáltuk majd photoshopban átszerkesztve, átszínezve képeket használtuk fel. A logo ötlete a név után jött és valami egyszerű, de mégis a névhez kapcsolódót szerettünk volna. Eleinte gondolkodtunk, hogy a Bootstrap vagy a Tailwind CSS lenne a mi esetünkben jó választás, de végül a weboldal alap design-ját Tailwind CSS-el formáltuk és alkottuk olyanra amelyet elképzeltünk. Azért lett ez a végső választás mivel könnyű használni és sokkal több mindenre nyújt lehetőséget, valamint ezt kevesebbet használtuk tanórákon és szerettünk volna egy kis kihívást is.

6. Összefoglalás

6.1. A szakdolgozat célja

A szakdolgozatunk célja egy olyan weboldal létrehozása volt, ami újra alkotja a használatú piacot és nem csak egy bizonyos nyelven, hanem a lehető legnagyobb lefedettséggel, akár az egész világon tudják használni a weboldalt. Az is a céljaink között volt, hogy egy könnyen használható és mindenki számára átlátható oldalt hozzunk létre. A felhasználók érdekeit előtérbe helyezve egy egyszerű és gyors weboldal elkészítése volt a célunk.

6.2. Megvalósítása

A fejlesztés során sok problémába ütköztünk, ami alatt nagyon sokat tanultunk. Sok olyan megoldást kellett használnunk, amit addig sosem. Nem volt könnyű a fejlesztés, de a csapatmunka nagyon könnyen ment és így áthidaltuk a nehézségeket.

Miután elkészültünk a design terveivel nekiálltunk az adatbázist is megtervezni. Amikor mindez elkészült nekiálltunk a fejlesztésnek. A fejlesztés során a design és az adatbázis is változott. Nagyon sokat segített, hogy volt egy kiindulási alap már amikor elkezdtünk programozni. Az adatbázis kulcsfontosságú volt az adatok tárolásában és kezelésében.

A bejelentkezés, valamint regisztráció biztonságossága nagyon fontos volt és elengedhetetlen. A titkosítás elmaradhatatlan volt.

A főoldal és a keretrendszer kialakításakor az volt a tervünk, hogy egy átlátható és könnyen használható navigációt hozzunk létre az oldalon. Ennek megvalósításában a React komponensek nagyon sokat segítettek és elengedhetetlen szerepet játszottak.

A homepage-ek kialakításánál a kinézet és a design követése elengedhetetlen volt. Bár változott a design a kezdetekhez képest, de a fő tematika megmaradt. Fő szempont a felhasználhatóság és átláthatóság volt. Minden a felhasználók igényei szerint készült.

A projekt legnehezebb része az új autó feltöltésének megoldása volt mivel ez igényelte a legtöbb időt és itt kellett a legtöbb dologra figyelni. Itt van a legtöbb interakciós lehetőség a felhasználók és a szerver között. Minden lehetséges hibára figyelmet kellett fordítanunk.

A backend fejlesztés során minden API hívást egy külön az adott oldal nevével ellátott mappában hoztunk létre. Ezen felül helyet kapott egy API mappa, ahol minden adatot le

lehet kérni és meg lehet tekinteni. Az adatok pontos lekérése, felküldése és kezelése volt a legnehezebb feladat mind közül.

A keresés megvalósítása is nagy kihívás volt mivel nagyon sok lehetséges kulcsszóra és keresési típusra oda kellett figyelni mely a tesztelés során nyújtott a legnagyobb problémát. A felhasználók profilja szintén sok kommunikációval jár a backend és frontend között így ennek megvalósítása sem volt a legkönnyebb.

A fejlesztés során nagyon sok új tapasztalatot és megoldási lehetőséget tanultunk amelyeket a jövőben tudunk majd hasznosítani.

7. Fejlesztési lehetőségek

7.1. Több nyelv

Az alap koncepció a világ minden táján lehetséges használat volt viszont ez nem teljesen tudott megvalósulni.

7.2. Több fajta jármű

Nem csak autókat, hanem egyéb járműveket is lehetne hozzáadni.

7.3. Adatok kezelése

A bejelentkezéskor egy elfelejtett jelszó esetén értesítés vagy e-mail, vagy telefon segítségével.

E-mailes hitelesítő regisztráció és bejelentkezés esetén.

8. Irodalomjegyzék

Fénykép források:

- <https://www.freepik.com/>

Tervezéshez és szerkesztéshez:

- <https://www.figma.com/>

Programozási problémák megoldásai:

- <https://stackoverflow.com>
- <https://chat.openai.com/>
- <https://react.dev/reference/react>
- <https://www.w3schools.com/nodejs/default.asp>

Ikonok:

- <https://www.flaticon.com/>

Használt segédletek:

- <https://tailwindcss.com/>
- <https://code.visualstudio.com/>
- <https://www.postman.com/>
- <https://laragon.org/>
- <https://react.dev/>
-

9. Mellékletek

9.1. Components példák

9.1.1. Keresés ár szerint

```
<label htmlFor="">Ár</label><br></br>
<input type="number" onChange={(e) => {handleInputChange(e)}} id="minprice"
name="min_price" className="border w-[180px] lg:w-[90px] py-3 px-4 text-
gray-700 focus:outline-none focus:ring-2 focus:ring-blue-300 rounded-sm
inline-block" placeholder="-tól" />
<input type="number" onChange={(e) => {handleInputChange(e)}} id="maxprice"
name="max_price" className="border w-[180px] lg:w-[90px] py-3 px-4 text-
gray-700 focus:outline-none focus:ring-2 focus:ring-blue-300 rounded-sm
inline-block" placeholder="-ig" />
```

9.1.2. Keresés meghaltatás szerint

```
<label htmlFor="">Kerékmeghajtas</label><br></br>
<input type="checkbox" name="wheels.fwd" id="wheels.fwd" onChange={(e) =>
{handleCheckbox(e)}}></input>
<input type="checkbox" name="wheels.rwd" id="wheels.rwd" onChange={(e) =>
{handleCheckbox(e)}}></input>
<input type="checkbox" name="wheels.awd" id="wheels.awd" onChange={(e) =>
{handleCheckbox(e)}}></input>
<input type="checkbox" name="wheels.4wd" id="wheels.4wd" onChange={(e) =>
{handleCheckbox(e)}}></input>
```

9.1.3. Teljesítmény megadása feltöltéskor

```
<div className="col-1 row-4 mb-3">
<h1 className="display: inline">Teljesítmény </h1><h1 className="display:
inline font-semibold">(1e)</h1>
<hr className="w-40 h-px bg-slate-400 border-0" />
<div className="w-10">
<input type="number" className="mt-1 w-32 border-2 border-gray-400 rounded-
lg cursor-pointer dark:text-gray-400" />
</div></div>
```

9.1.4. Kép feltöltés

```
<h1 className="font-bold mt-5">Kép Feltöltése</h1>
<hr className="w-full h-px bg-slate-400 border-0" />
<div className="w-92 mt-2 place-self-center">
```

```
<input className="block w-full text-sm text-gray-900 border-2 border-gray-400 rounded-lg cursor-pointer bg-gray-50 dark:text-gray-400 focus:outline-none dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400" id="file_input" type="file" /></div>
```

9.1.5. Ár megadása feltöltéskor

```
<div className="col-span-2 row-4">
  <h1 className="display: inline">Kívánt ár </h1><h1 className="display inline font-semibold">(Ft)</h1>
  <hr className="w-full h-px bg-slate-400 border-0" />
  <div className="w-fit">
    <input type="text" className="mt-1 ml-5 w-[312px] border-2 border-gray-400 rounded-lg cursor-pointer dark:text-gray-400" required />
  </div></div>
```

9.1.6. Feltöltés limitáló

```
const [limitReached, setLimitReached] = React.useState(false);
useEffect(() => {
  axios.get("/api/my_profile?include_unlisted_cars=true").then((res)
=> {
    if(((parseInt(res.data["data"]["permissions"]) >>> 2) & 1) == 0)
    {
      // nincs végtelen feltöltése
      console.log(Object.keys(res.data["data"]["car"]));
      if(Object.values(res.data["data"]["car"]).length > 5) {
        setLimitReached(true);
      }
    }
  })
}, [setLimitReached]);
```