

Győri SZC Jedlik Ányos Gépipari és Informatikai

Technikum és Kollégium

FORD SALON

Autó Kereskedés webalkalmazás

Dernóczi Dániel – Hujber Patrik Tamás

Győr, 2023

**Záródolgozat feladatkiírás**

|  |  |
| --- | --- |
| Tanulók nevei: | Dernóczi Dániel, Hujber Patrik Tamás |
| Képzés: | nappali |
| Szak: | 5 0613 12 03 Szoftverfejlesztő és -tesztelő technikus |

**A záródolgozat címe:**

**Ford Salon**

Konzulens: Bólya Gábor

Beadási határidő: 2023. 04. 29

Győr, 2023. 10. 01

**Módos Gábor** igazgató

**Konzultációs lap**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | A konzultáció | Konzulens aláírása |
| ideje | témája |
| 1. | 2021.10.01. | Témaválasztás és specifikáció |  |
| 2. | 2022.02.01. | Záródolgozat készültségi fokának értékelése |  |
| 3. | 2022.03.17. | Dokumentáció véglegesítése |  |

*Konzultációs lap.*

**Tulajdonosi nyilatkozat**

Ez a dolgozat a saját munkánk eredménye. Dolgozatunk azon részeit, melyeket más szerzők munkájából vettük át, egyértelműen megjelöltük.

Ha kiderülne, hogy ez a nyilatkozat valótlan, tudomásul vesszük, hogy a szakmai vizsgabizottság a szakmai vizsgáról minket kizár és szakmai vizsgát csak új záródolgozat készítése után tehetünk.

Győr, 2023. 04. 27

Tanulók aláírásai:

Dernóczi Dániel

Hujber Patrik Tamás

# **Tartalomjegyzék**

Tartalom

[1. **Tartalomjegyzék** 4](#_Toc133517277)

[2. **Bevezetés** 6](#_Toc133517278)

[2.1. **Téma bemutatása** 6](#_Toc133517279)

[2.2. **A tervezése** 7](#_Toc133517280)

[2.2.1. **Reszponzivitás** 7](#_Toc133517281)

[2.2.2. **Animációk** 8](#_Toc133517282)

[2.3. **A jövő** 8](#_Toc133517283)

[2.4. **A csapatmunka** 9](#_Toc133517284)

[2.5. **Publikálás** 10](#_Toc133517285)

[3. **A program** 11](#_Toc133517286)

[3.1. **Technológiák** 11](#_Toc133517287)

[3.1.1. **Frontend – Vue.js** 11](#_Toc133517288)

[3.1.2. **Backend – Node.js** 13](#_Toc133517289)

[3.1.3. **Adatbázis – MY-SQL** 15](#_Toc133517290)

[3.2. **Az adatbázis felépítése** 16](#_Toc133517291)

[3.2.1. **Táblák** 17](#_Toc133517292)

[3.3. **Jelszavak** 19](#_Toc133517293)

[3.4. **Jogosultságok** 19](#_Toc133517294)

[3.5. **Tesztelés** 20](#_Toc133517295)

[4. **Felhasználói kézikönyv** 21](#_Toc133517296)

[4.1. **Webes és Mobile nézet** 21](#_Toc133517297)

[4.2. **Kezdőlap** 22](#_Toc133517298)

[4.3. **Navigációs menü** 22](#_Toc133517299)

[4.4. **Footer** 22](#_Toc133517300)

[4.5. **Modellválaszték** 23](#_Toc133517301)

[4.6. **Konfigurátor** 24](#_Toc133517302)

[4.7. **Rendelés összegzése (Konfigurátoron belül)** 27](#_Toc133517303)

[4.8. **Regisztráció** 28](#_Toc133517304)

[4.9. **Bejelentkezés** 28](#_Toc133517305)

[4.10. **Profil oldal** 29](#_Toc133517306)

[4.11. **Egyéb oldalak** 29](#_Toc133517307)

[4.11.1. **Hírek** 29](#_Toc133517308)

[4.11.2. **Rólunk** 29](#_Toc133517309)

[4.11.3. **Története** 29](#_Toc133517310)

[4.11.4. **Források** 30](#_Toc133517311)

[5. **Saját konklúzió a feladat készítéséről** 30](#_Toc133517312)

# **Bevezetés**

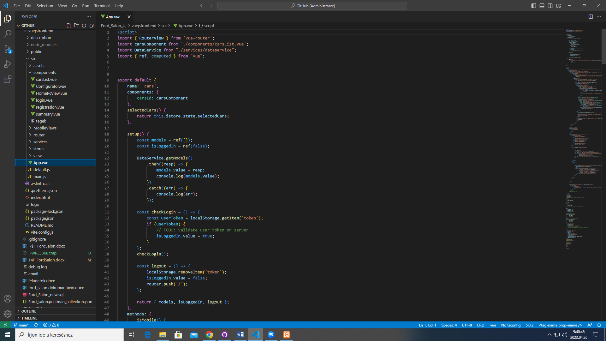
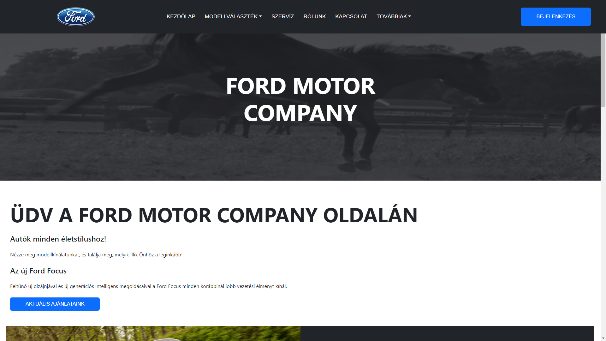
## **Téma bemutatása**

A Ford Salon autó konfigurátor weboldal egy olyan platform, amely lehetővé teszi a felhasználók számára, hogy testre szabják saját Ford járművüket. A weboldal Vue.js, Node.js és MySQL technológiákkal készült, amelyek biztosítják a gyors és hatékony működést, valamint a megbízható adatkezelést.

A felhasználók az oldalon kiválaszthatják az autó típusát, majd az autó és a karosszéria színét és az extrákat is, amelyek az autó kialakítása és a személyes preferenciák kielégítése érdekében személyre szabható.

Az oldal felhasználóbarát felülettel rendelkezik, amely lehetővé teszi a felhasználók számára, hogy könnyedén navigáljanak a különböző lehetőségek között.

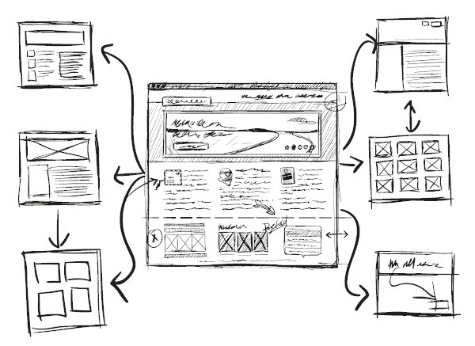
Összességében a Ford Salon autó konfigurátor weboldal egy hatékony és megbízható platform, amely lehetővé teszi a felhasználók számára, hogy egyszerűen testre szabják saját Ford autójukat. Az oldal intuitív felépítése, valamint biztonságos adatkezelése miatt ideális választás az autó testre szabására vágyó ügyfelek számára.



Kezdő lap

App.vue kód

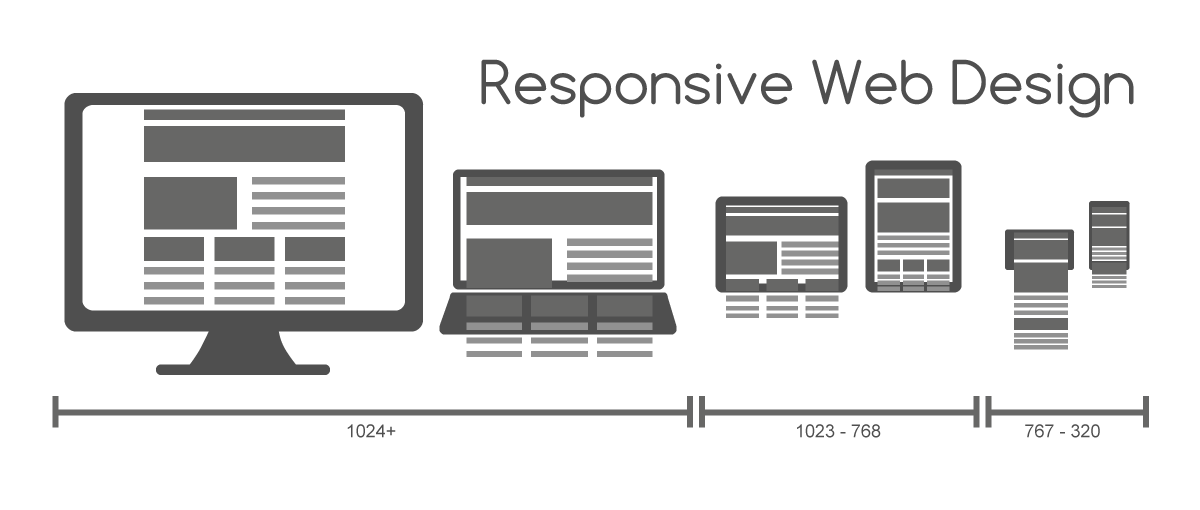
## **A tervezése**

Az oldal tervezése a felhasználói élményre és a funkcionalitásra összpontosított. Az oldal dizájnja modern és letisztult, ugyanakkor a színek és a grafikák segítik a felhasználókat abban, hogy könnyen tájékozódjanak és használják az oldalt.

Tervezés illusztráció

A tervezés során fontos szempont volt az oldal könnyen kezelhetősége is. Az oldal célja az autók értékesítése, ezért a funkciókat és a felületeket úgy alakítottuk ki, hogy egyszerűen és gyorsan lehessen azokat használni. A felhasználók könnyedén böngészhetnek az autók között, konfigurálhatják azokat és meg is rendelhetik az oldalon.

### **Reszponzívitás**

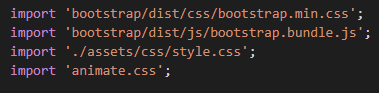
****Az oldal tervezésénél figyelembe vettük a mobilbarát kialakítást is, hogy a felhasználók bármilyen eszközről és helyről könnyedén elérhessék az oldalt és használhassák az összes funkciót. Ezért egy reszponzív design-t készítettünk, amely automatikusan alkalmazkodik az adott eszközhöz és mérethez, így az oldal mindig optimális formában jelenik meg.

Reszponzívitás illusztráció

illusztráció

### **Animációk**

Az animációk a weboldal vizuális megjelenésének dinamikusabbá tételét szolgálják, emellett javítják a felhasználói élményt és segítenek a felhasználók figyelmének felkeltésében. Az oldalon különböző animációk találhatóak, például az oldal megnyitásakor és az autók listázásakor. Az animációk kiválasztása során fontos szempont volt a megfelelő sebesség és átláthatóság, valamint a felhasználói élmény javítása. A Vue.js keretrendszer animáció támogatást biztosít, ami lehetővé tette számunkra, hogy könnyen integráljuk az animációkat a weboldalunkba.



Animáció importálása

## **A jövő**

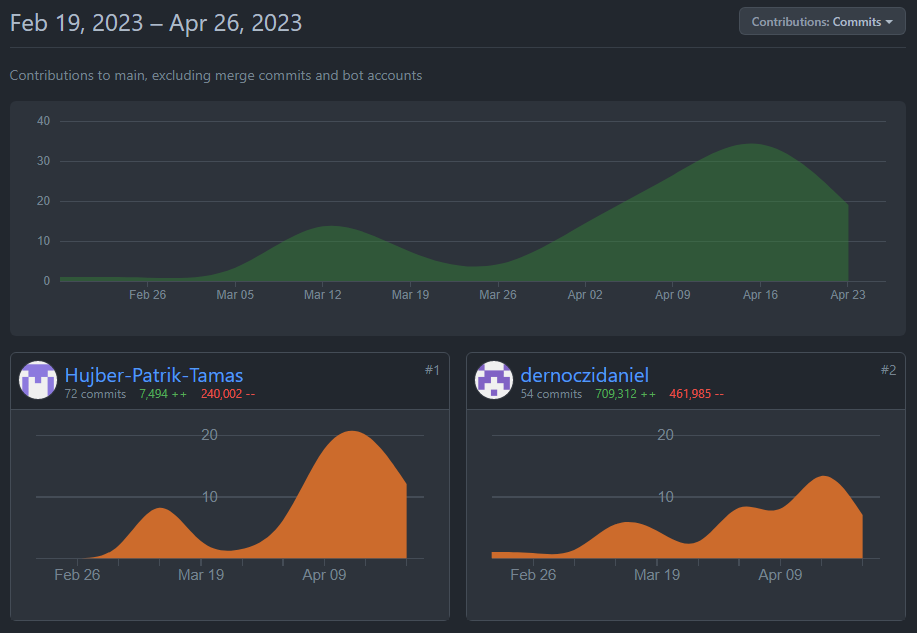
Az oldal további fejlesztéseinek célja, hogy a felhasználói élményt és az oldal funkcionalitását még jobbá tegyük. Néhány tervezett fejlesztés:

* **Szervízelés online** – A jövőben szeretnénk hozzá adni az eredetileg is tervezett szervízelés lehetőségét, csak az idő és segítség hiányából fakadóan, ezt nem tudtuk kivitelezni.
* **Mobil applikáció** – A jövőben tervezzük egy mobil alkalmazás kifejlesztését, amely lehetővé teszi az ügyfeleink számára, hogy könnyedén és kényelmesen böngésszenek az autóink között és konfigurálják azokat bárhol és bármikor.
* **Tesztvezetések lehetősége** - Az oldalunk további fejlesztése közé tartozik a tesztvezetések lehetősége, amely lehetővé teszi a felhasználóknak, hogy igényeljenek egy tesztvezetést és meggyőződjenek a választásuk helyességéről.

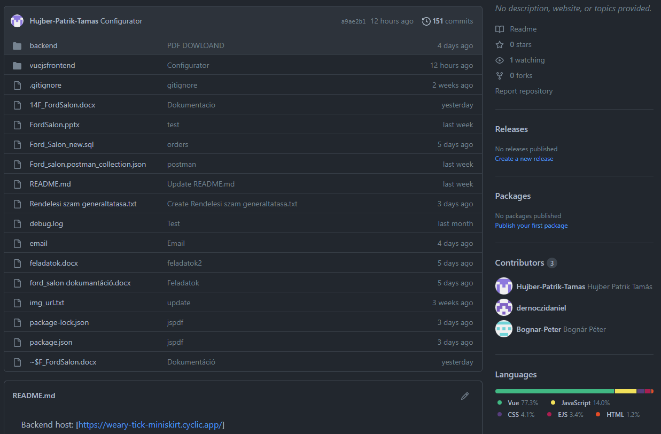
Ezen fejlesztésekkel az oldalunkkal még jobban ki tudnánk szolgálni a felhasználóink igényeit, és jobb felhasználói élményt nyújtani számukra.

## **A csapatmunka**

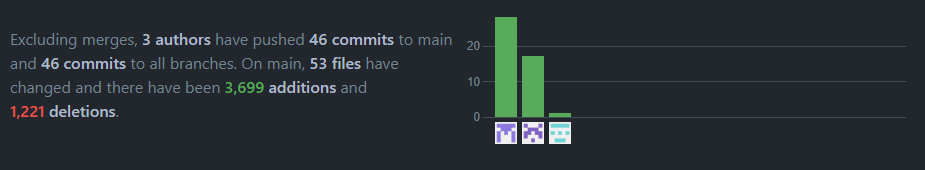
A weboldal készítésén ketten dolgoztunk, ezért fontos volt jól felosztani a feladatokat.

A hatékonyság érdekében az egyikünk a Front-End-del foglalkozott, míg a másikunk a Back-End-ért és az Adatbázisért volt felelős elsősorban. Azonban mindketten programoztunk mindenhol.

GitHub logó

A GitHubot használtuk a verziókezelés és a kód megosztása céljából. Rendszeresen feltöltöttük a kódjainkat a GitHub-ra, ahol mindkettőnk számára elérhető volt a kód. Ezt a módszert használtuk arra is, hogy ellenőrizzük és átnézzük egymás kódjait, és javítsuk a hibákat.

Contributors



Pulse

Code

## **Publikálás**

Az adatbázist futtatását az iskolai Linux szerver végzi, Kubernetes segítségével. A backendet a GitHub segítségével.

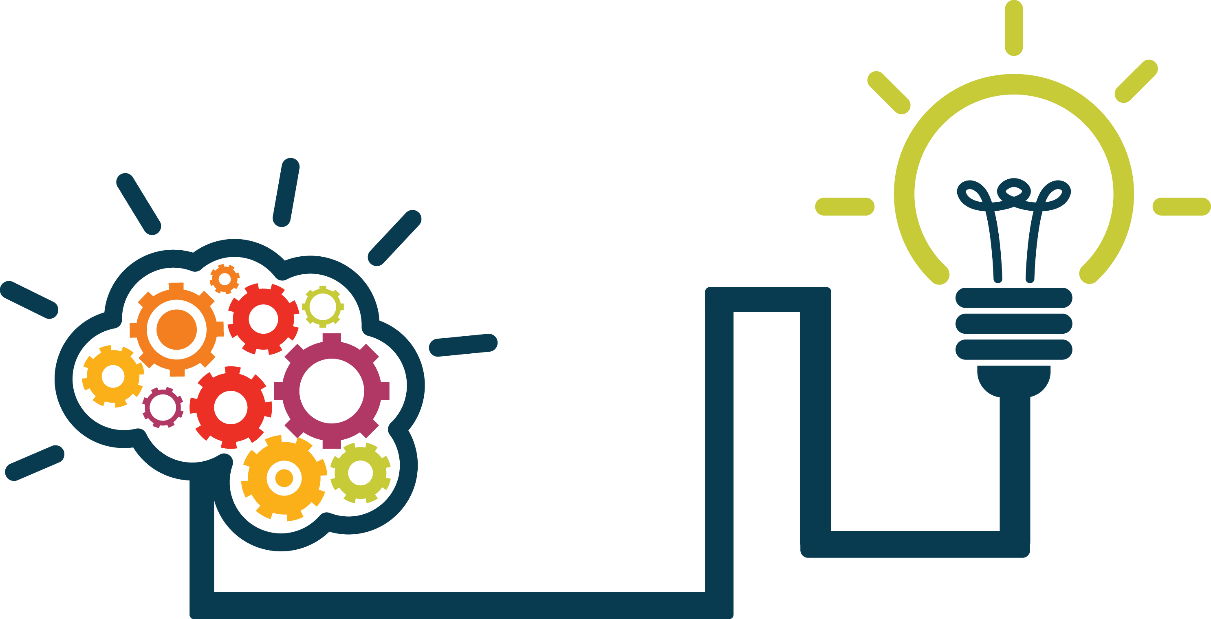
A publikálással szeretnénk gyorsan használhatóvá tenni a weboldalunkat. A szerveren a weboldalunk jól fut az optimalizálás miatt.



Böngészők logó

## **Problémák és megoldások**

A szakdolgozat késztése közben akadtak néhol problémák, mint például a VueJs-ben a mobil nézettel, amit egy mobileView segítségével oldottunk meg. Az oldalak közti adat átvitelnél is akadtunk néhány nehézségbe, de sikerült megoldani ezt a fennakadást. Az backend készítésé során is voltak kisebb nagyobb fennakadások, amiket utána nézéssel, olvasással sikeresen meg tudtunk csinálni.



# **A program**

## **Technológiák**

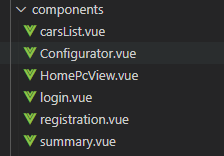
### **Front-End – Vue.js**

A Vue.js-t választottuk, mert a tanulmányaink alatt erről a nyelvről tanultunk a legtöbbet, mellesleg nagyon megtetszett számunkra a felépítése, használata és hatékonysága, amilyen könnyen tudja kezelni az adatkötéseket, így a felhasználói felületünk gyorsabb és interaktívabb lett.

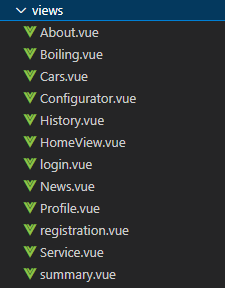
Vue.js logó

A kódszintaxisa nagyon olvasható és érthető, ami megkönnyíti a csapatmunkát, fejlesztést és a kód karbantartását. Emellett a Vue.js rendelkezik egy nagyon aktív közösséggel, így rengeteg kiegészítő, bővítmény, és oktatóanyag elérhető hozzá, így a tudásunkat is ki tudtuk egészíteni azzal, hogy utána olvastunk az interneten a különböző problémák megoldásának, erre sokszor volt szükségünk, mert rengeteg mindent nem tanultunk az iskola keretei között, aminek fő oka az idő hiánya volt. Ez nagyon hasznos volt az oldalunk fejlesztése során.

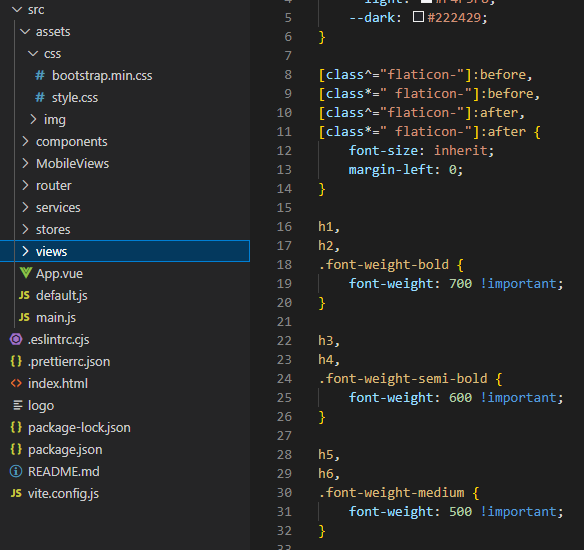
A Front-End a Vue.js keretrendszerre épül és az alábbi főbb részekből áll:

* **Komponensek:** A weboldal egyedi felépítésű és stílusú komponensekkel rendelkezik, amelyek egyszerűen újrahasznosíthatók és testre szabhatók a különböző oldalakon és funkciókban.

Komponensek

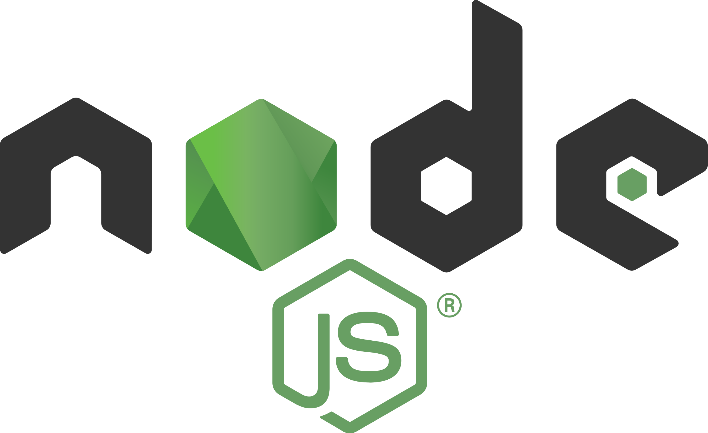
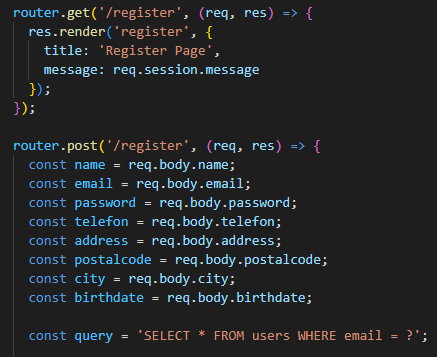
* **Views:** A weboldal egyedi felépítésű és stílusú Views-el rendelkezik, amelyek meghívják a komponenseket.
* **Routing:** Az oldal a Vue Router modult használja a dinamikus útvonalak kezeléséhez, ami lehetővé teszi az egyszerű és intuitív navigációt az oldalon.
* **Állapotkezelés:** A Vuex állapotkezelő modul segítségével történik az állapotkezelés az oldalon, amely lehetővé teszi a komponensek közötti adatmegosztást és a könnyű állapotfrissítést.

Views

* **Kéréskezelés:** Az oldal a Vue Resource és a Axios kéréskezelő modulokat használja az API hívásokhoz és az adatok lekérdezéséhez a Back-End-ről.
* **Stílus:** CSS stílusokat használtunk a weboldal dizájnolásához.

Style.css

### **Back-End – Node.js**

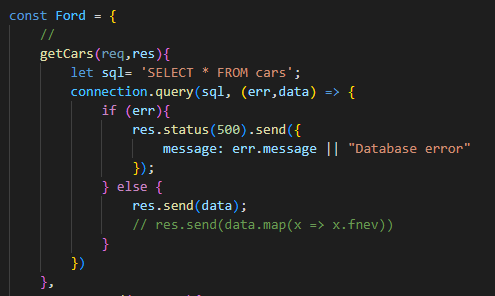
A Node.js-t választottuk a Back-Endfejlesztéséhez, mert nagyon hatékonyan kezeli az esemény vezérelt programozást, ami különösen fontos a nagy mennyiségű adatfeldolgozás és a szinkronizálás kezelése során, mellesleg tanulmányaink alatt leginkább ebből a nyelvből éreztük, hogy képesek lennénk a projekt elkészítésére. Emellett a Node.js nagyon jól skálázódig, így képes nagy terhelések kezelésére is. A JavaScript nyelvet használja, ami segített egységes kódolási szabványokat tartani a Front-End és a Back-Endközött. A kódolás során tartottuk magunkat a tiszta kód szabályához. A Node.js-t választottuk továbbá azért is, mert számos bővítmény és könyvtár elérhető hozzá, amelyek megkönnyítik a fejlesztést és lehetővé teszik az erőforrások hatékonyabb kihasználását.

Node.js logó

A Node.js Back-End rész nagyon fontos szerepet játszik az oldal működésében. Az alábbiakban részletezem a használt technológiákat és azok felépítését:

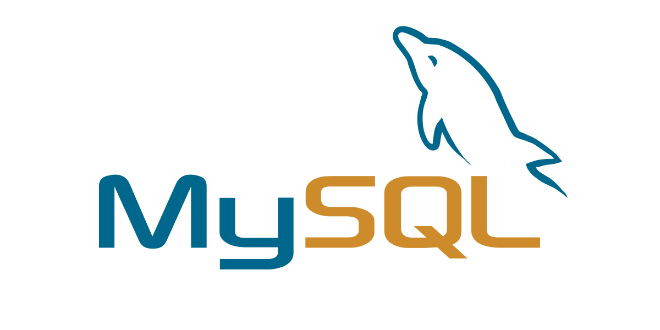
Példa a tiszta kódra

* **Express.js:** Az Express.js egy Node.js alapú webalkalmazás-keretrendszer, amely lehetővé teszi a webalkalmazások gyors és hatékony készítését. A Back-End-ben az Express.js-t használtuk a webalkalmazásunk kiszolgálására.
* **Passport:** A Passport egy Middleware, amely autentikációs szolgáltatásokat biztosít Node.js alkalmazásokhoz. A Back-End-ben használtuk a Passport-ot a felhasználók bejelentkeztetéséhez és az azonosításuk ellenőrzéséhez.
* **JWT (JSON Web Tokens):** A JWT egy JSON-alapú token rendszer, amelyet gyakran használnak az internetes alkalmazásokban a felhasználói hitelesítésre. A Back-End-ben használtuk a JWT-t a felhasználói munkamenetek kezelésére és az autentikációra.
* **bcrypt:** A bcrypt egy jelszó-hashelő algoritmus, amelyet a felhasználók jelszavának biztonságos tárolására használtunk.
* **Dotenv:** A Dotenv egy környezeti változó kezelő modul, amely lehetővé teszi, hogy a fejlesztők titkos adatokat tárolhassanak, például adatbázis-kapcsolati adatokat, jelszavakat, kulcsokat és más konfigurációs adatokat.

Ezen technológiák és modulok összekapcsolásával hoztuk létre a Node.js Back-End részét, amely lehetővé teszi az adatok tárolását és az autentikációt, valamint a kommunikációt a Front-End-el.

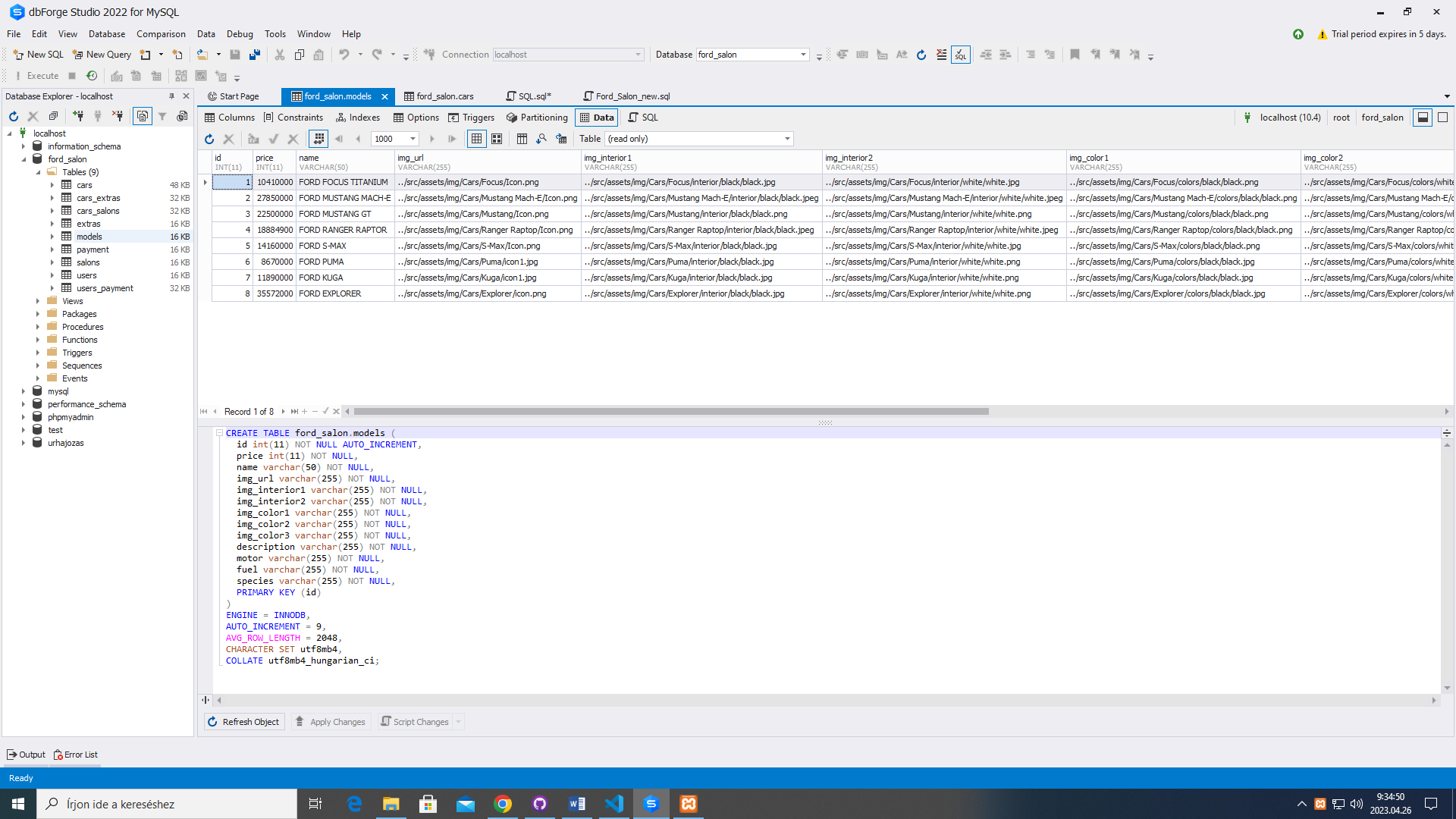
Példa a get metódusra

### **Adatbázis – MySQL**

****Az adatbázisunkhoz a MySQL-t választottuk, mert ez egy megbízható és skálázható relációs adatbázis-kezelő rendszer.

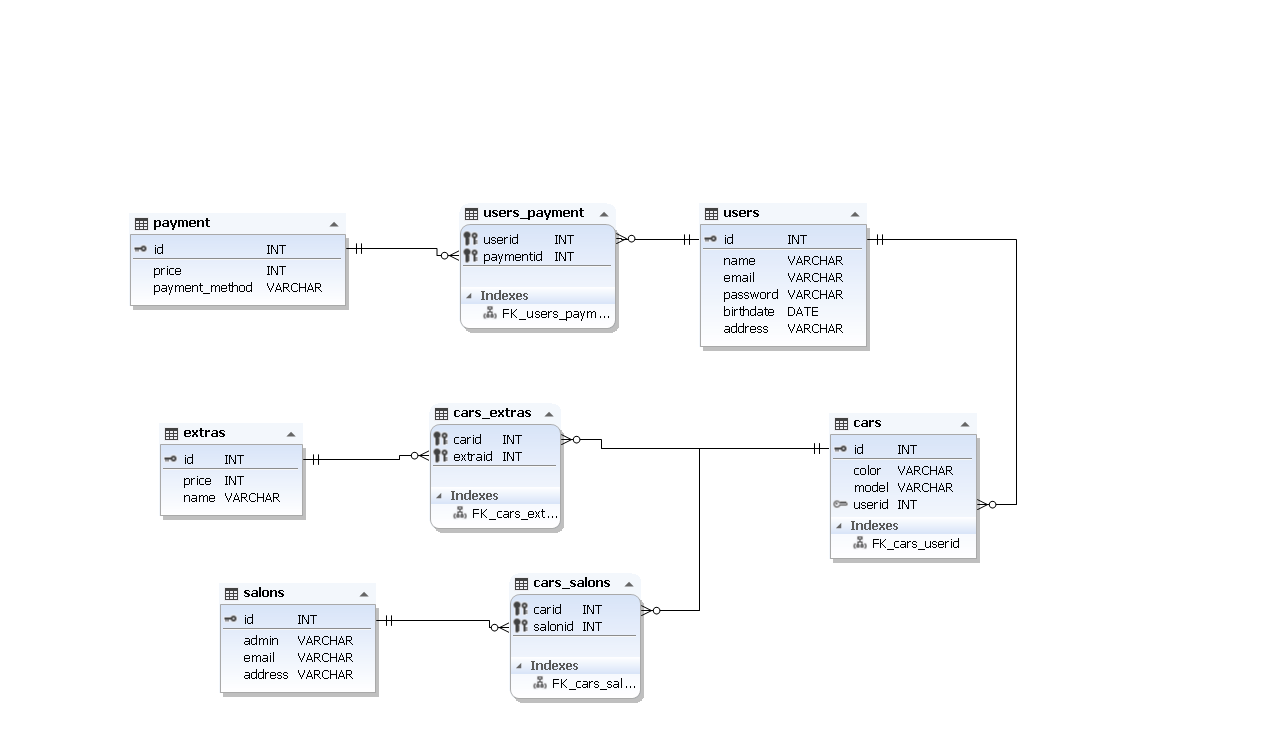
MySQL logó

Az SQL viszonylag egy egyszerűbb és érthetőbb nyelv, ami számunkra a használat alatt nagyon megtetszett. A MySQL gyors és hatékony adatbázis-motorral rendelkezik, ami fontos volt az oldalunk nagy mennyiségű adatának kezelése során. Emellett az adatok egyszerűen és hatékonyan szervezhetők, és lehetőséget nyújt az adatbázisok közötti adatkapcsolatok kezelésére. A MySQL-t széles körben használják, így sok tapasztalat és támogatás érhető el hozzá, ami segíti a fejlesztést és a karbantartást.



Példa a táblákra

## **Az adatbázis felépítése**



Az adatbázis felépítése azért fontos, mert az hatással van az adatok tárolására, kezelésére és elérhetőségére. Az adatok hatékony tárolása és könnyű elérése alapvető fontosságú az üzleti és személyes alkalmazások számára, mivel ezek az adatok lehetővé teszik az adatok kezelését, elemzését és felhasználását. Az adatbázis felépítése befolyásolja az adatok integritását, biztonságát és hozzáférhetőségét is.

Példa az adatbázis felépítéséről

Az adatbázis felépítése magában foglalja az adatok tárolási módjának és formátumának kiválasztását, az adatok struktúrájának meghatározását, az adatok tárolási helyének kiválasztását, az adatok hozzáférhetőségének biztosítását, az adatok redundanciájának minimalizálását és az adatok biztonságának garantálását.

### **Táblák**

A felsorolásban láthatók az adatbázisban szereplő fontosabb táblák és kapcsolótáblák:

**cars:** Az autók adatait tartalmazó tábla. Minden sor egy autó adatait tartalmazza, például az autó típusát, modelljét, évjáratát, motorját és árát.

**extras:** Az extrák adatait tartalmazó tábla. Minden sor egy extrát tartalmaz, például navigációs rendszert, bőrüléseket vagy ülésfűtést.

**price:** Az autók árát tartalmazó tábla.

**salons:** Az autószalonok adatait tartalmazó tábla. Minden sor egy autószalon adatait tartalmazza, például a nevét, címét, és elérhetőségeit.

**users:** A felhasználók adatait tartalmazó tábla. Minden sor egy felhasználó adatait tartalmazza, például a nevét, email címét és jelszavát.

A táblák közötti kapcsolatokat több kapcsolótábla segítségével valósítottuk meg. Az alábbiakban felsoroljuk ezeket:

**cars\_extras:** Az autók és extrák kapcsolatát tartalmazó tábla. Minden sor egy autó és egy extra kapcsolatát tartalmazza.

**cars\_salons:** Az autók és autószalonok kapcsolatát tartalmazó tábla. Minden sor egy autó és egy autószalon kapcsolatát tartalmazza.

Ezek a kapcsolótáblák lehetővé teszik, hogy az alkalmazás az adatbázisban tárolt információkat összekapcsolja és lehetővé tegye például, hogy egy felhasználó válasszon egy autót és kiválassza, hogy melyik autószalonban veheti át.

A **cars** tábla a következő oszlopokat tartalmazza:

**id:** Az autó egyedi azonosítója, amely egy egész szám típusú érték.

**userid:** Az autó tulajdonosának azonosítója, amely egy egész szám típusú érték. Ez egy külső kulcs, amely hivatkozik a users tábla id oszlopára.

**model:** Az autó modellje, amely egy szöveges típusú érték.

**motor:** Az autó motorja, amely egy szöveges típusú érték.

**color:** Az autó karosszériájának színe, amely egy szöveges típusú érték.

**interiorcolor:** Az autó belső terének színe, amely egy szöveges típusú érték.

Ezenkívül a kapcsolótáblákban szereplő **cars\_extras** és **cars\_salons** táblák is hivatkoznak az id oszlopra, amely az autók egyedi azonosítója. Ezek a kapcsolótáblák teszik lehetővé, hogy az alkalmazás az adatbázisban tárolt információkat összekapcsolja és lehetővé tegye például, hogy egy felhasználó válasszon egy autót, majd kiválassza a kívánt extra opciókat.

A **users** tábla az alkalmazás felhasználóinak adatait tárolja. Az oszlopok a következők:

**id:** Az egyedi azonosítója a felhasználónak, amely egy egész szám típusú érték.

**name:** A felhasználó teljes neve, amely egy szöveges típusú érték.

**email:** A felhasználó E-Mail címe, amely egy szöveges típusú érték.

**password:** A felhasználó jelszava, amely egy szöveges típusú érték. Az adatbázisban tárolt jelszavakat általában hash-elve tárolják, hogy biztonságosabbak legyenek.

**birthdate:** A felhasználó születési dátuma, amely egy dátum típusú érték.

**address:** A felhasználó címe, amely egy szöveges típusú érték.

Az id oszlop az egyedi azonosító, amelyet a kapcsolatok létrehozásához használunk más táblákkal, például a cars vagy a users\_payment táblákkal. Az email és a password oszlopokat azonosításra és hitelesítésre használjuk a felhasználó bejelentkezésekor. Az address oszlop a felhasználó címét tartalmazza, amely például szállítási információk megadására hasznos. Az birthdate oszlop pedig a felhasználó születési dátumát tartalmazza, amely egyéb funkciókban, például az életkor kiszámításában is felhasználható.

## **Jelszavak**

Ha sikeresen regisztrál a felhasználó, akkor azután a Back-End-re érkező jelszó titkosítására kerül sor, melyet ebben a formában tárolunk el az adatbázisban. A lekérések során ez a titkosított jelszó semmiféle formában nem kerül visszaküldésre a Front-End irányába az esetleges támadások végett. A felhasználó által megadott titkosítatlan jelszó nem kerül tárolásra az adatbázisban.

## **Jogosultságok**

A felhasználói jogosultság azonosítja, hogy az adatbázisban tárolt adatokhoz milyen szintű hozzáférése van az egyes felhasználóknak. A jogosultságok szintje általában a felhasználók szerepe szerint van meghatározva és biztosítja, hogy csak azok a felhasználók férjenek hozzá az adatokhoz, akiknek erre szükségük van a feladatuk ellátásához.

Az adatbázisban tárolt adatok hozzáférési szintjeiből nálunk kettő van:

* **User:** Ez a jogosultsági szint engedélyezi a felhasználóknak, hogy az adatbázisban tárolt adatokat megtekinthessék, olvashassák, és azokat lekérdezhessék.
* **Admin:** Ez a jogosultsági szint engedélyezi a felhasználóknak, hogy az adatbázis struktúráját, az adatbázis rendszerbeállításait és az adatbázis-kezelő rendszert felügyeljék és módosítsák.

## **Tesztelés**

A Postman egy olyan tesztelő eszköz, amely lehetővé teszi a fejlesztők számára, hogy teszteljék és ellenőrizzék az API-kat és a webszolgáltatásokat. A Postman segítségével lehetőség van az API-k automatizált tesztelésére, dokumentálására és megosztására is.

Postman logó

Az API tesztelés során a Postman lehetővé teszi az HTTP kérések manuális vagy automatizált végrehajtását. A fejlesztők létrehozhatnak kéréseket, amelyek tartalmazzák az API-khoz szükséges adatokat, például a paramétereket, a fejléceket és az autentikációs adatokat. Ezután a Postman lehetővé teszi az ilyen kérések gyors és hatékony végrehajtását.

Az API teszteléséhez a Postman lehetővé teszi a különböző teszt típusok végrehajtását. Például lehetőség van a funkcionális tesztek elvégzésére, amelyek ellenőrzik, hogy az API-k megfelelően működnek-e a tervezett funkcionalitás tekintetében. Ezen felül a Postman lehetővé teszi a terhelési tesztek végrehajtását is, amelyek ellenőrzik, hogy az API-k megfelelően működnek-e a nagy terhelések alatt is.

A Postman tesztelő eszköz lehetővé teszi a teszt eredményeinek rögzítését és dokumentálását is, amelyek segítségével a fejlesztők könnyen visszakövethetik a tesztelési folyamatot és az eredményeket. Ezen felül a Postman lehetővé teszi az API-k dokumentálását is, amely segít az API-k használatának megértésében és dokumentálásában.

# **Felhasználói kézikönyv**

## **Webes és Mobil nézet**

Az oldalunk reszponzív tervezéssel készült, így képes alkalmazkodni a különböző kijelző méretekhez és eszközökhöz. Ez azt jelenti, hogy az oldal megjelenése és felépítése automatikusan változik attól függően, hogy milyen eszközt használsz.

A webes nézet a számítógépek és laptopok számára optimalizált. Ebben a nézetben az oldal teljes funkcionalitását és tervezését megtekintheted, beleértve az autók listáját, az egyéni profilodat, az autó konfigurátor funkciót és minden egyéb opciót, amelyet az oldalunk kínál.

A mobil nézet a kisebb kijelzővel rendelkező eszközökre optimalizált, például okostelefonokra vagy tabletekre. Ebben a nézetben az oldal tartalma és tervezése átalakul, hogy a felhasználók kényelmesen és hatékonyan használhassák az oldalt a kisebb képernyőn. Az autók listája, az egyéni profilod és az autó konfigurátor funkció szintén elérhető a mobil nézetben.

Mobil nézet

Az oldalunk reszponzív tervezése biztosítja, hogy a felhasználók bármilyen eszközön élvezhessék az oldalunk nyújtotta élményt, és könnyedén használhassák a funkciókat és opciókat a kijelző méretétől és típusától függetlenül.

## **Kezdőlap**

A Ford Salonnak a kezdőlapján hirdetések és az új autókkal kapcsolatos információk láthatók, hogy a felhasználó könnyedén megtalálja a számára megfelelő autót.

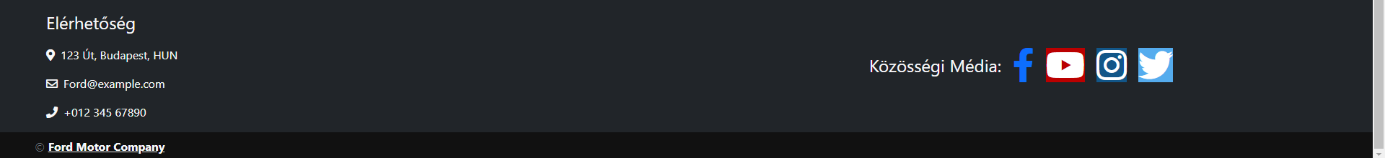
Kezdőlap

## **Navigációs menü**

Az oldal tetején található menü segítségével a felhasználók könnyen navigálhatnak az oldalon és hozzáférhetnek az oldal többi részéhez, például az autók listájához vagy a bejelentkezéshez.

## **Footer**

Az oldal alján található footerben találhatók a fontos linkek, az elérhetőségek és a közösségi média ikonjai, amelyek segítségével a felhasználók könnyen kapcsolatba léphetnek velünk.

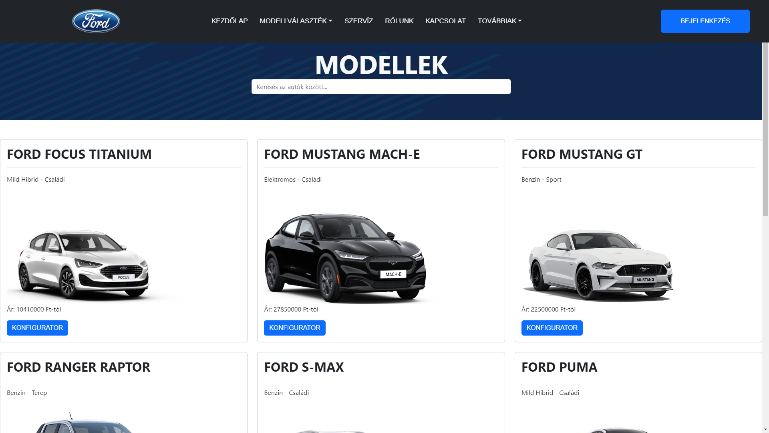


Footer

## **Modellválaszték**

A modellválaszték egy különálló oldalon jelenik meg, ahol az összes rendelkezésre álló Ford modellt meg lehet tekinteni, majd közülük a felhasználók választhatnak, hogy melyiket szeretnék konfigurálni. Az oldal a következő elemekből áll:

* **Keresőmező**: Az oldal tetején található egy keresőmező, amely lehetővé teszi a felhasználók számára, hogy könnyen megtalálják a keresett modellt a nevének megadásával.
* **Modell lista**: Az oldal közepén található modell lista bemutatja az összes rendelkezésre álló Ford modellt. A modellek képpel, névvel és árral vannak ellátva, hogy a felhasználók könnyen kiválaszthassák a megfelelő modellt.

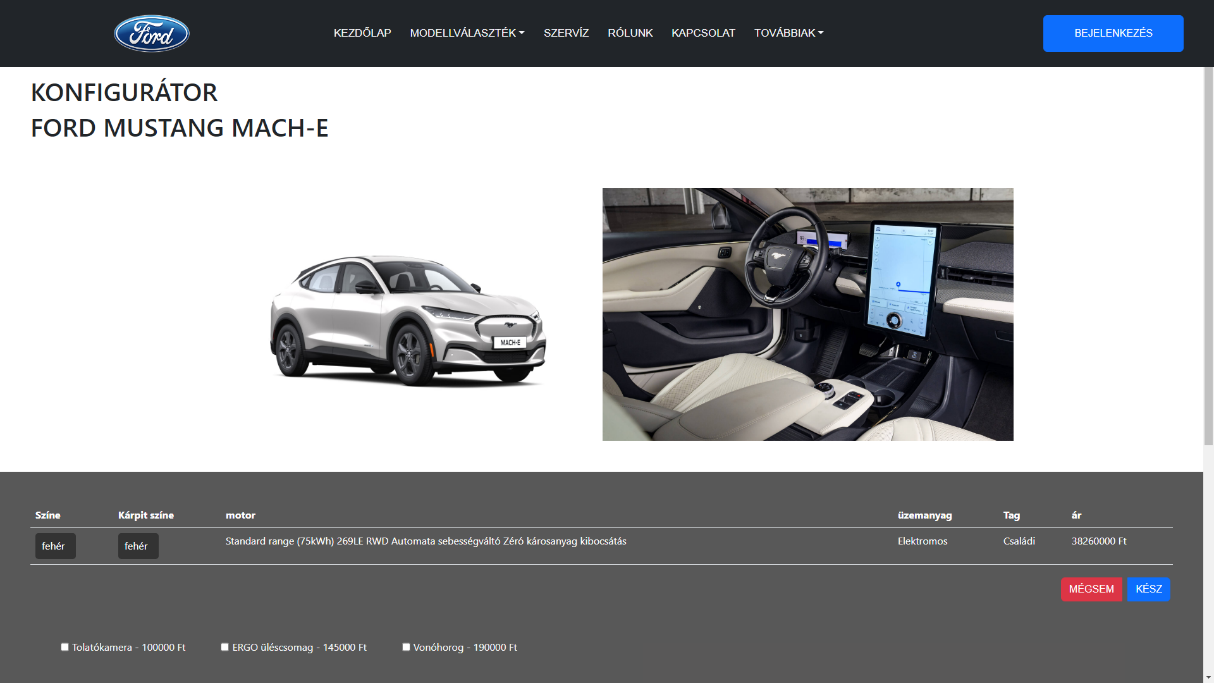


Modellválaszték oldal

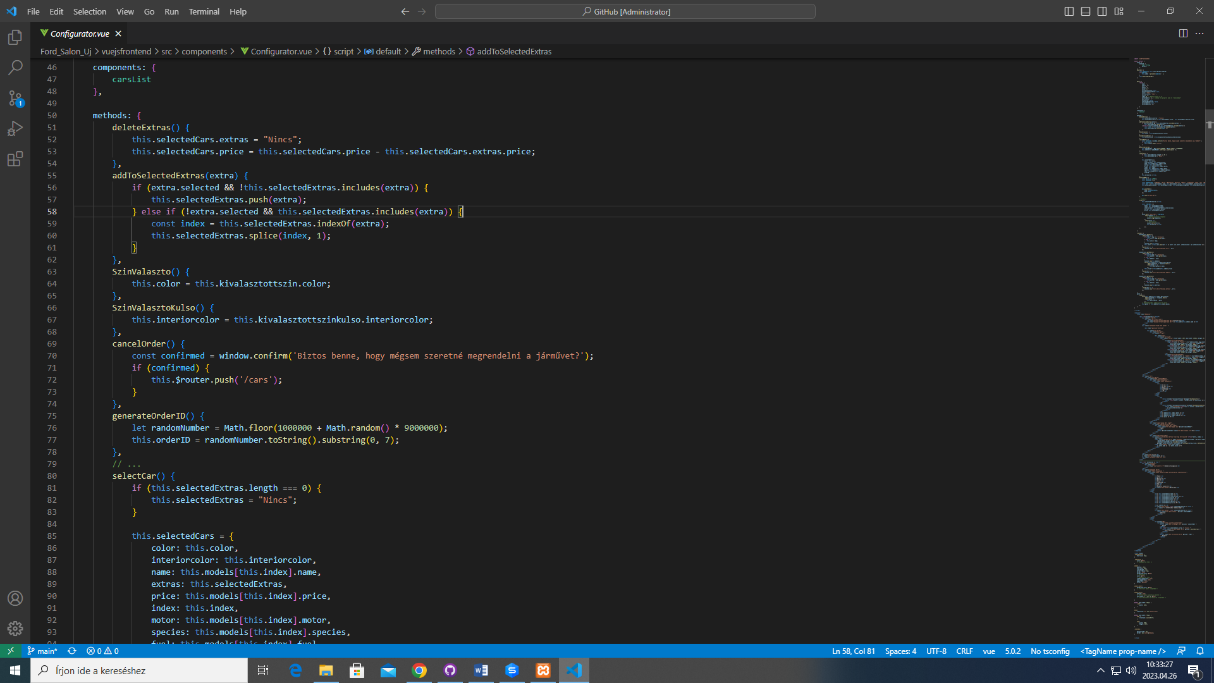
## **Konfigurátor**

Az oldal különböző autó modelleket, azok tulajdonságait és árait listázza. Az oldal fő eleme egy interaktív autó konfigurátor, ami Vue.js keretrendszerben íródott, ez teszi lehetővé a felhasználóknak, hogy kiválaszthatják a modellt, a színt, az opcionális kiegészítőket és sok más funkciót, ezzel egyidőben pedig megtekinthetik a kiválasztott konfiguráció árát. A felhasználók ezzel a funkcióval pedig személyes preferenciáik szerint tudják testre szabni választott autójukat.

Emellett pedig sok más információt is nyújt a konfigurátor, mint például a motor fajtája és üzemanyag szükséglete.



Konfigurátor oldal



Konfigurátor kód részlet

**Komponensek**

Az oldal egy fő komponensből áll, amely a következőket tartalmazza:

* carsList komponens: Az autómodellek listáját jeleníti meg.
* Props-ok
* modelId: Az aktuálisan kiválasztott autómodell azonosítója.
* selectedCar: Az aktuálisan kiválasztott autómodellek listája.
* Adattagok
* name: Az autó neve.
* models: Az elérhető autómodellek listája.
* cars: Az elérhető autók listája.
* extras: Az elérhető extra felszerelések listája.
* price: Az autó ára.
* color: Az autó külső színe.
* interiorcolor: Az autó belső színe.
* colors: Az elérhető külső és belső színek listája.
* index: Az aktuálisan kiválasztott autómodell indexe a listában.
* selectedCars: Az aktuálisan kiválasztott autómodellek listája.
* description: Az autómodell leírása.
* Computed properties
* color: A külső szín getter-je.

**Metódusok (pl.)**

* SzinValaszto(): A külső szín változtatásáért felelős metódus.
* SzinValasztoKulso(): A belső szín változtatásáért felelős metódus.
* Send(): Az autó színeinek beállításáért felelős metódus.
* cancelOrder(): Az autórendelés visszavonásáért felelős metódus.

**Életciklus metódusok**

* mounted(): A komponens mountolása után az aktuálisan kiválasztott autómodell indexe inicializálódik.

**Watcherek**

* index(): Az aktuálisan kiválasztott autómodell indexe megváltozik. Ez után a leírás és az ár beállítása történik.

**Szolgáltatások**

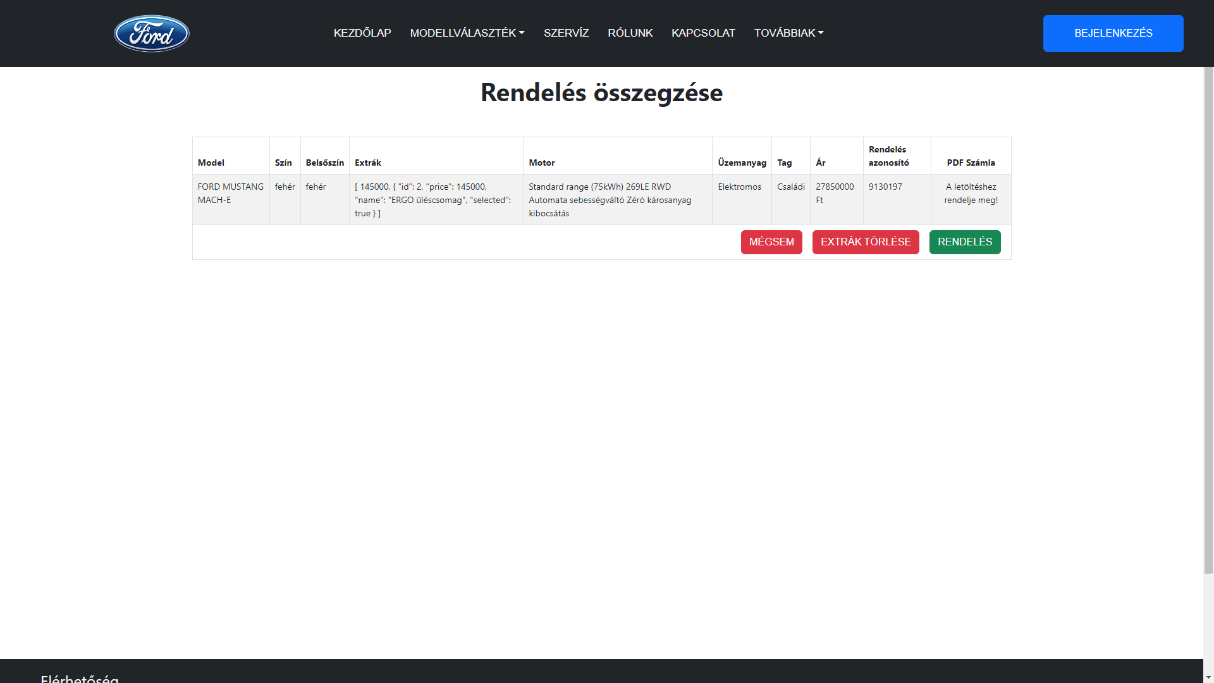
Az oldal a következő szolgáltatásokat használja:

* **DataService.getCars():** Az elérhető autók lekérdezéséhez szükséges.
* **DataService.getModels():** Az elérhető autómodellek lekérdezéséhez szükséges. DataService.bookCar(carId, startDate, endDate): Az autó foglalásához szükséges szolgáltatás, amely a megadott autó azonosító, valamint a foglalás kezdő és végdátumának paramétereit várja.
* **DataService.cancelBooking(bookingId):** A már meglévő foglalások törléséhez szükséges szolgáltatás, amely a törlendő foglalás azonosítóját várja paraméterként.

## **Rendelés összegzése (Konfigurátoron belül)**

A rendelés összegzése oldal azon folyamat végpontja, amikor a felhasználó elkészítette a kívánt autó konfigurációját és véglegesíti a rendelést. Az oldal megjelenése a felhasználó által választott autó konfigurációjának összegzését tartalmazza, amely magában foglalja a választott modellt, a kiegészítőket, az árat és a kiválasztott szolgáltatásokat. Az oldalon a felhasználó kiválaszthatja a fizetési módot, valamint a szállítási címet, és megadhatja az összes szükséges információt a rendelés elküldéséhez.

Az oldal funkcionalitásának fejlesztése során a Vue.js keretrendszer és a Node.js környezet együttesen alkalmazva biztosította a megfelelő szintű adatkezelést és az összes szükséges funkció megvalósítását. A MySQL adatbázis segítségével pedig biztosítva volt az adatok tárolása és lekérdezése a rendelés összegzése oldalon. Az oldal kialakítása és a felhasználói élmény javítása érdekében az animációk is fontos szerepet játszottak az oldalon.



Rendelés

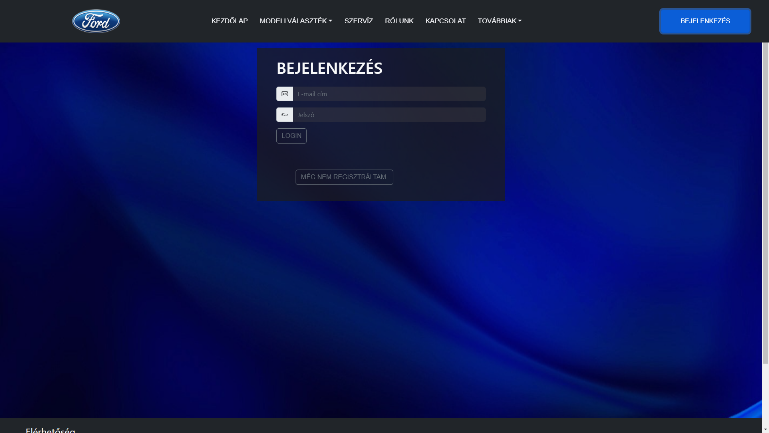
## **Regisztráció**

Az oldalon található regisztrációs űrlap lehetőséget biztosít az új felhasználók számára, hogy regisztráljanak és létrehozzanak egy fiókot. A regisztrációhoz szükséges információkat (pl. felhasználónév, E-Mail, jelszó stb.) meg kell adni az űrlapon, majd a "Regisztráció" gombra kattintva elküldhetőek az adatok a szervernek.

Regisztráció

## **Bejelentkezés**

Az oldal bejelentkezési űrlapján a felhasználóknak meg kell adniuk a Regisztráció során megadott E-Mail címüket és jelszavukat. Miután megadták ezeket az adatokat, rákattintanak a "Bejelentkezés" gombra, és a rendszer ellenőrzi a bejelentkezési adatokat.



Bejelentkezés

## **Profil oldal**

## **Egyéb oldalak**

### **Hírek**

Az oldalunkon található egy "Hírek" oldal, ahol a legfrissebb autókkal kapcsolatos hírek és információk tekinthetők meg. Az oldalunkon rendszeresen frissítjük a híreket, így mindig naprakész információkról, a legújabb trendekről, autó modellekről és technológiáról lehet tudomást szerezni.

### **Rólunk**

Ezen az oldalon a Ford csapatáról és a vezetőről lehet információkat találni. Bemutatja, milyen értékeket képvisel a Ford, milyen célokat tűzött ki maga elé és hogy kívánja elérni ezeket.

### **Története**

A "Története" menüpontja egy átfogó történeti áttekintést nyújt a Ford vállalat születésétől napjainkig.

Bemutatjuk, hogyan kezdődött az egész, milyen körülmények között jött létre az első Ford modell, és hogyan alakult ki az autógyártásban betöltött szerepe. Az oldalon megtalálhatóak azok a fontos mérföldkövek, amik az évek során történtek, mint például az első összeszerelő sor megjelenése, az ikonikus Model T bemutatása vagy az autógyártás terén elért innovációk.

A "Története" menüpontban nem csak az autógyártásra fókuszálunk, hanem bemutatjuk a Ford vállalat átható értékeit és szemléletét is. Elmondjuk, milyen elvek vezérlik a munkát, milyen szerepet játszik a fenntarthatóság a vállalat tevékenységeiben és hogy hogyan igyekszik hozzájárulni a társadalom fejlődéséhez.

### **Források**

A források oldalon lehetőség van azoknak az anyagoknak a megtekintésére, amelyek felhasználásra kerültek a weboldal elkészítése során. Ide tartoznak a képek, ikonok, szöveges anyagok, stb. Az oldal célja az átláthatóság és a forrásmegjelölés, valamint a jogi követelményeknek való megfelelés. A források között szerepelnek külső linkek, amelyek az adott anyagok forrásához vezetnek, valamint a weboldal belső tartalmai, amelyekhez hivatkozással lehet megtekinteni a használt anyagokat.

# **Saját konklúzió a feladat készítéséről**

A szakdolgozat készítése közben sokat tanultunk a csoportban való dolgozásról, arról hogy a kitűzött tervekből mi valósítható meg, és ezáltal még jobban megismerkedtünk a használt programozási nyelvekkel. A segítséget leginkább az interneten található különböző oldalak nyújtották. A feladat készítése közben rá kellet jönnünk, hogy kutató munka nélkül nem fogunk tudni haladni, mert az iskola keretein belül sok mindent nem érintettünk, amire szükségünk lett volna.