

Készítette: Gerengay Krisztián, Kruzslicz Ádám

BMSZC Verebély László Technikum

Szoftverfejlesztő és –tesztelő technikus képzés

2025. szeptember

Tartalomjegyzék

[Bevezetés 5](#_Toc208838213)

[Felhasználói dokumentáció 6](#_Toc208838214)

[Regisztráció 7](#_Toc208838215)

[Bejelentkezés és kijelentkezés 8](#_Toc208838216)

[Hirdetések böngészése és szűrése 8](#_Toc208838217)

[Hirdetés feltöltése 9](#_Toc208838218)

[Saját hirdetések kezelése 9](#_Toc208838219)

[Kedvencek kezelése 10](#_Toc208838220)

[Üzenetküldés 10](#_Toc208838221)

[Profil szerkesztése 11](#_Toc208838222)

[Backend – Szerveroldali technológiák 12](#_Toc208838223)

[Frontend – Felhasználói felület 12](#_Toc208838224)

[Adatbázis és adattárolás 13](#_Toc208838225)

[Egyéb technológiák 14](#_Toc208838226)

[Rendszerkövetelmények 14](#_Toc208838227)

[Felhasználói oldal (böngésző) 14](#_Toc208838228)

[Szerveroldal (üzemeltetés) 15](#_Toc208838229)

[Egyéb 15](#_Toc208838230)

[Adminisztrációs felület 16](#_Toc208838231)

[Funkciók 16](#_Toc208838232)

[Biztonság és jogosultságok 18](#_Toc208838233)

[Felhasználói élmény 18](#_Toc208838234)

[Telepítési útmutató 18](#_Toc208838235)

[Előfeltételek 18](#_Toc208838236)

[Forráskód letöltése 18](#_Toc208838237)

[Függőségek telepítése 19](#_Toc208838238)

[Környezeti változók beállítása 19](#_Toc208838239)

[Adatbázis létrehozása XAMPP-pal 19](#_Toc208838240)

[Alkalmazás kulcs generálása 19](#_Toc208838241)

[Adatbázis migrációk futtatása 19](#_Toc208838242)

[Tárhely linkelése a képekhez 19](#_Toc208838243)

[Az alkalmazás futtatása XAMPP-pal 20](#_Toc208838244)

[Jogosultságok Windows alatt 20](#_Toc208838245)

[Adatbázis struktúra 20](#_Toc208838246)

[Főbb táblák és mezők 20](#_Toc208838247)

[Kapcsolatok 22](#_Toc208838248)

[Frontend megvalósítás 22](#_Toc208838249)

[Blade sablonmotor 23](#_Toc208838250)

[Bootstrap 23](#_Toc208838251)

[Interaktív elemek – JavaScript és AJAX 23](#_Toc208838252)

[Felhasználói nézetek 24](#_Toc208838253)

[Ikonok és vizuális elemek 24](#_Toc208838254)

[Hibakezelés és visszajelzés 24](#_Toc208838255)

[Backend megvalósítás 25](#_Toc208838256)

[Modellek és adatbázis 25](#_Toc208838257)

[Vezérlők (Controllers) 25](#_Toc208838258)

[Validáció és hibakezelés 25](#_Toc208838259)

[Jogosultságkezelés és biztonság 26](#_Toc208838260)

[RESTful útvonalak és API-k 26](#_Toc208838261)

[Fájlkezelés 26](#_Toc208838262)

[Üzenetküldés és kapcsolatok 26](#_Toc208838263)

[Tesztelés és hibakeresés 26](#_Toc208838264)

[Biztonság 26](#_Toc208838265)

[Felhasználói hitelesítés és jogosultságkezelés 27](#_Toc208838266)

[Jelszavak védelme 27](#_Toc208838267)

[Adatvalidáció és hibakezelés 27](#_Toc208838268)

[CSRF védelem 27](#_Toc208838269)

[SQL injection elleni védelem 27](#_Toc208838270)

[Fájlkezelés és jogosultságok 27](#_Toc208838271)

[XSS elleni védelem 27](#_Toc208838272)

[Hozzáférés-szabályozás 28](#_Toc208838273)

[Naplózás és hibakövetés 28](#_Toc208838274)

[Tesztelés 28](#_Toc208838275)

[Automatizált tesztek 28](#_Toc208838276)

[Tesztelési folyamat 28](#_Toc208838277)

[Tesztelési területek 29](#_Toc208838278)

[Manuális tesztelés 29](#_Toc208838279)

[Hibakezelés és naplózás 29](#_Toc208838280)

[Továbbfejlesztési lehetőségek 29](#_Toc208838281)

[Függelék 31](#_Toc208838282)

[Főbb modellek és kapcsolataik 31](#_Toc208838283)

[Fontosabb útvonalak (routes) 32](#_Toc208838284)

[Validációs szabályok példák 32](#_Toc208838285)

[Hibakezelés és visszajelzések 32](#_Toc208838286)

[Használt csomagok, technológiák 33](#_Toc208838287)

[Példa adatbázis-struktúra (részlet) 33](#_Toc208838288)

[Tesztelés 33](#_Toc208838289)

[Hasznos parancsok fejlesztéshez 33](#_Toc208838290)

[Elérhetőségek, támogatás 34](#_Toc208838291)

[Ábrajegyzék 35](#_Toc208838292)

# Bevezetés

A digitális világ fejlődésével az online autókereskedés egyre nagyobb teret hódít, hiszen a vásárlók és eladók számára is kényelmes, gyors és átlátható megoldást kínál. A CarHub Autókereskedés weboldal egy modern, felhasználóbarát platform, amely a használt autók adásvételét, hirdetését és keresését teszi lehetővé, miközben számos kényelmi funkcióval támogatja a felhasználókat. Az alábbiakban részletesen bemutatom az oldal működését, főbb funkcióit, valamint a felhasználói élményt.

A CarHub főoldalára érkezve a látogató egy letisztult, áttekinthető felülettel találkozik, ahol az aktuális autóhirdetések böngészése azonnal elérhető. A hirdetések között szűrőkkel lehet keresni, például márka, modell, évjárat, ár vagy üzemanyag-típus szerint. Ez a funkció különösen hasznos, hiszen a felhasználó pillanatok alatt megtalálhatja az igényeinek megfelelő járműveket, anélkül, hogy hosszasan kellene lapozgatnia a találatok között. A hirdetések kártyaszerű megjelenítése, a képek, árak és főbb adatok kiemelése segíti a gyors áttekintést.

A regisztráció folyamata egyszerű és gyors: a felhasználónak meg kell adnia egy felhasználónevet, e-mail címet, jelszót, valamint egy rövid bemutatkozást. A rendszer ellenőrzi az adatok helyességét, például a jelszó hosszát vagy az e-mail cím formátumát. A regisztráció után a felhasználó bejelentkezhet, és elérhetővé válnak számára a személyre szabott funkciók, mint például a hirdetésfeltöltés, saját hirdetések kezelése, kedvencek mentése vagy üzenetküldés.

Az autóhirdetés feltöltése során a felhasználó egy részletes űrlapot tölt ki, ahol megadhatja a jármű minden fontos adatát: márka, modell, évjárat, ár, futott kilométer, üzemanyag-típus, váltó, szín, karosszéria, extrák, valamint részletes leírás. A képek feltöltése is támogatott természetesen: a rendszer ellenőrzi a képek formátumát és méretét, így elkerülhetők a hibás vagy túl nagy fájlok. A feltöltött hirdetés azonnal megjelenik a főoldalon, így a többi felhasználó is böngészheti azt.

A saját hirdetések menüpontban a felhasználó áttekintheti, szerkesztheti vagy törölheti korábbi hirdetéseit. Ez a funkció lehetővé teszi, hogy a hirdetések mindig naprakészek legyenek, és a felhasználó könnyedén menedzselje az eladásra kínált autóit. A szerkesztés során módosíthatók az adatok, új képek tölthetők fel, vagy régiek törölhetők.

A CarHub egyik kiemelkedő funkciója a kedvencek kezelése. Böngészés közben bármelyik hirdetés elmenthető a kedvencek közé, így a felhasználó később könnyen visszatalálhat az őt érdeklő autókhoz. A kedvencek listája egy külön menüpontban érhető el, ahol a hirdetések egy kattintással el is távolíthatók.

Az üzenetküldő rendszer lehetőséget ad arra, hogy a vásárlók közvetlenül felvegyék a kapcsolatot az eladókkal, hiszen a hirdetések adatlapján keresztül üzenet küldhető az eladónak, aki a saját üzenetei között láthatja a beérkezett megkereséseket, és válaszolhat is rájuk. Az üzenetek két fülön jelennek meg: beérkezett és elküldött üzenetek, így a kommunikáció átlátható és könnyen követhető.

A márkák és modellek kezelése szintén fontos része az oldalnak. A felhasználók böngészhetnek a különböző autómárkák között, megtekinthetik az egyes márkákhoz tartozó modelleket, sőt, adminisztrátori jogosultsággal új márkát vagy modellt is hozzáadhatnak, illetve törölhetnek. Ez a funkció biztosítja, hogy a hirdetésfeltöltés során mindig naprakész, valós márka- és modellválaszték álljon rendelkezésre.

A profilkezelés lehetőséget ad a felhasználónak, hogy módosítsa személyes adatait, jelszavát vagy bemutatkozását. A biztonság érdekében a jelszavak titkosítva tárolódnak, és az e-mail címeket a rendszer nem adja ki harmadik félnek.

Összességében a CarHub Autókereskedés weboldal egy összetett, mégis könnyen használható platform, amely a használt autók adásvételének minden lépését támogatja. A felhasználóbarát felület, a részletes keresési és szűrési lehetőségek, a hirdetéskezelés, a kedvencek és az üzenetküldés mind hozzájárulnak ahhoz, hogy az autóvásárlás vagy -eladás élménye gyors, biztonságos és hatékony legyen. Az oldal kialakítása során kiemelt figyelmet kapott a biztonság, az adatvédelem és a felhasználói élmény, így mindenki megtalálhatja a számára legmegfelelőbb járművet, vagy könnyedén értékesítheti saját autóját.

A CarHub Autókereskedés weboldal egy modern, online platform, amely a használt autók adásvételét, hirdetését és keresését teszi egyszerűvé és átláthatóvá mind a vásárlók, mind az eladók számára. Az oldal célja, hogy biztonságos, felhasználóbarát környezetet biztosítson, ahol a felhasználók könnyedén böngészhetnek a hirdetések között, saját autókat tölthetnek fel, kapcsolatba léphetnek egymással, valamint kezelhetik kedvenceiket és profiljukat. A CarHub minden funkciója azt szolgálja, hogy az autóvásárlás és -eladás folyamata gyors, kényelmes és hatékony legyen.

A CarHub Autókereskedés weboldal fejlesztése során számos korszerű, iparági szabványnak megfelelő technológiát alkalmaztunk, hogy a felhasználók számára biztonságos, gyors és élvezetes élményt nyújtson a weboldal.

# Felhasználói dokumentáció

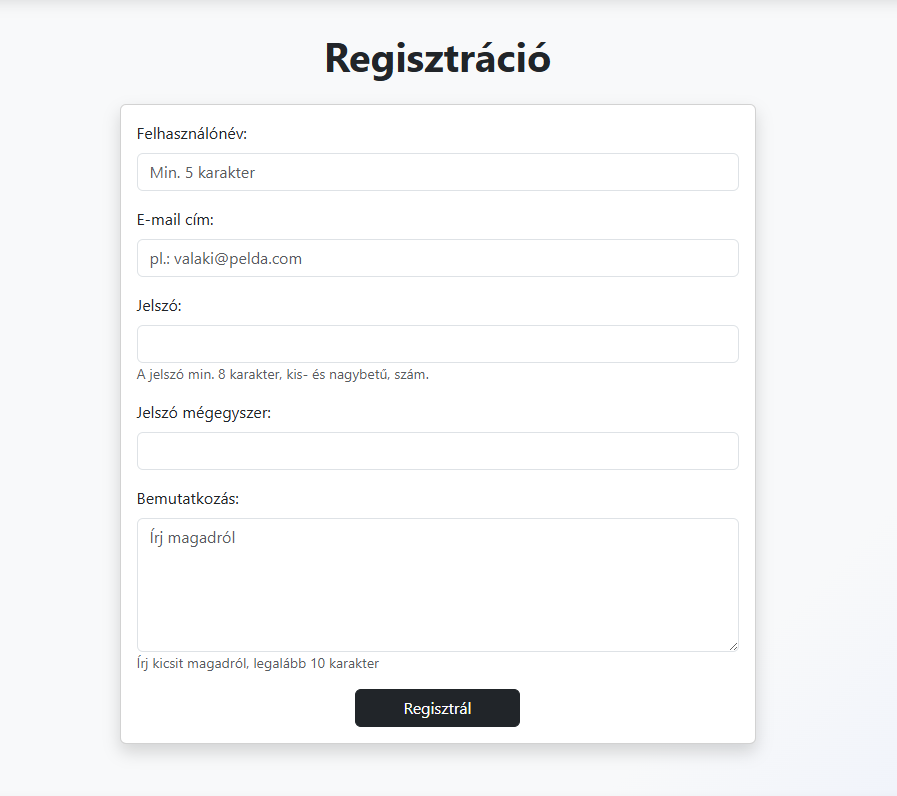
Ez a fejezet részletesen bemutatja a CarHub autókereskedés webalkalmazás használatát, lépésről lépésre végigvezetve a legfontosabb funkciókon.

## Regisztráció

- A főoldalon vagy a felső menüsorban található „Regisztráció” gombra kattintva éred el a regisztrációs űrlapot.

- Az űrlapon add meg a következő adatokat:

* Felhasználónév: Legalább 5 karakter hosszú, egyedi név.
* E-mail cím: Valós, működő e-mail cím.
* Jelszó és jelszó megerősítése: A jelszónak meg kell felelnie a biztonsági követelményeknek.
* Bemutatkozás: Legalább 10 karakteres rövid leírás magadról.

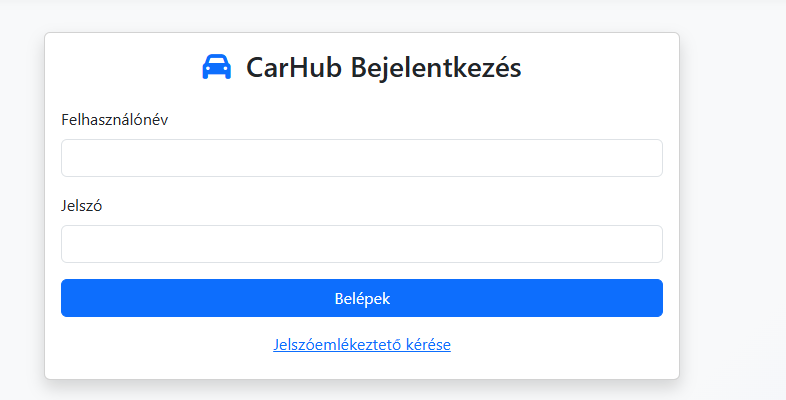


1. ábra Regisztráció

- Hibás vagy hiányos kitöltés esetén a rendszer figyelmeztet, és kiírja a hibákat.

- Sikeres regisztráció után automatikusan átirányít a bejelentkezési oldalra.

## Bejelentkezés és kijelentkezés

- A „Bejelentkezés” menüpontban add meg a regisztrált e-mail címedet és jelszavadat. 

2. ábra Bejelentkezés

- Sikeres bejelentkezés után a neved megjelenik a menüsorban, és elérhetővé válnak a hirdetésfeltöltési, szerkesztési, üzenetküldési funkciók.

- Elfelejtett jelszó esetén a „Jelszóemlékeztető” gombra kattintva kérhetsz új jelszót.

- A „Kijelentkezés” gombbal bármikor kiléphetsz a fiókodból.

## Hirdetések böngészése és szűrése

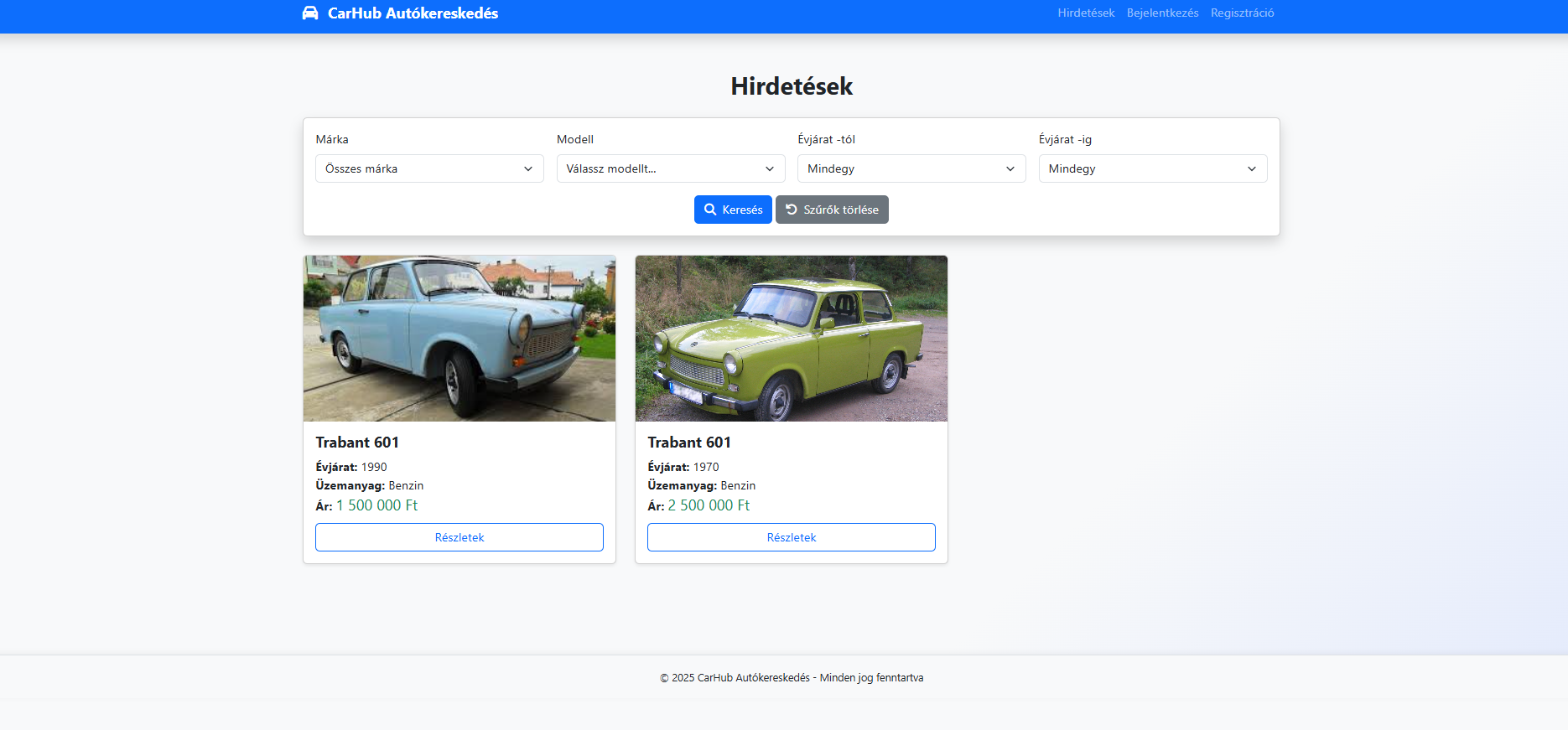
- A főoldalon vagy a „Hirdetések” menüpontban böngészheted az összes elérhető autóhirdetést.

- A lista felett szűrő űrlap található, ahol a következő paraméterek szerint szűrhetsz:

* Márka (pl. Opel, Ford)
* Modell (a márka kiválasztása után dinamikusan betöltődnek a modellek)
* Évjárat (tól-ig)
* Ár (tól-ig)
* Üzemanyag típus (benzin, dízel, hibrid, elektromos stb.)

- A szűrők beállítása után a „Keresés” gombra kattintva jelennek meg a találatok.

- Egy hirdetésre kattintva részletes adatlap nyílik meg, ahol képeket, leírást, extrákat, elérhetőséget láthatsz.



3. ábra Hirdetések megjelenítése

## Hirdetés feltöltése

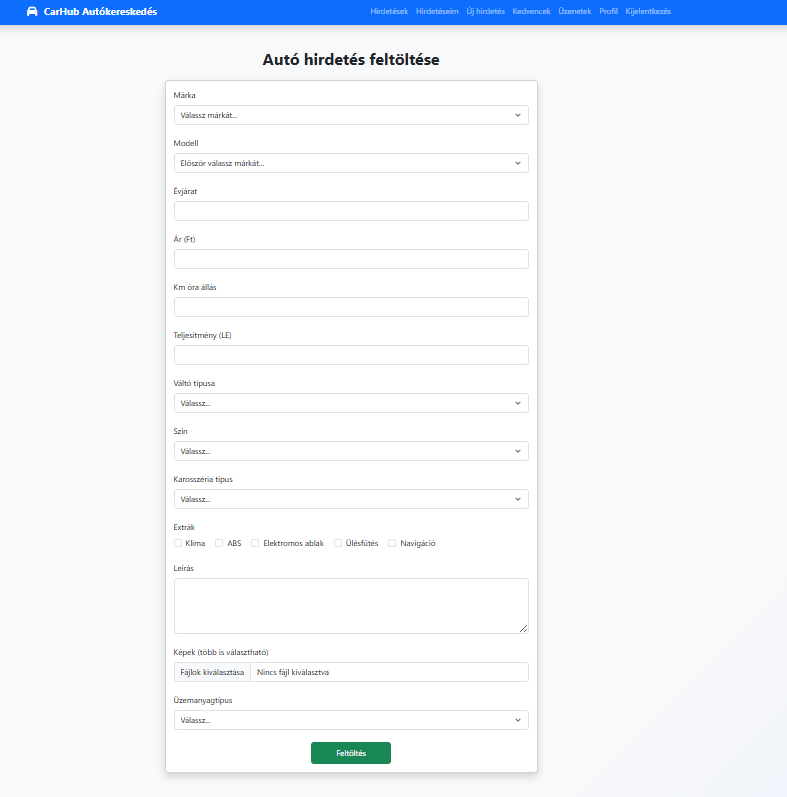
- Bejelentkezés után a „Hirdetés feltöltése” gombra kattintva új autóhirdetést adhatsz fel.

- Az űrlapon meg kell adni:

* Márka és modell (a modell lista dinamikusan töltődik a márka alapján)
* Évjárat, ár, kilométeróra állás, teljesítmény, üzemanyag, váltó, szín, karosszéria, extrák, leírás
* Képek feltöltése (több kép is feltölthető, csak képfájlok, max. 2MB/db)

- Hibás vagy hiányos kitöltés esetén a rendszer figyelmeztet.

- Sikeres mentés után a hirdetés azonnal megjelenik a listában.



4. ábra Új hirdetés feladása

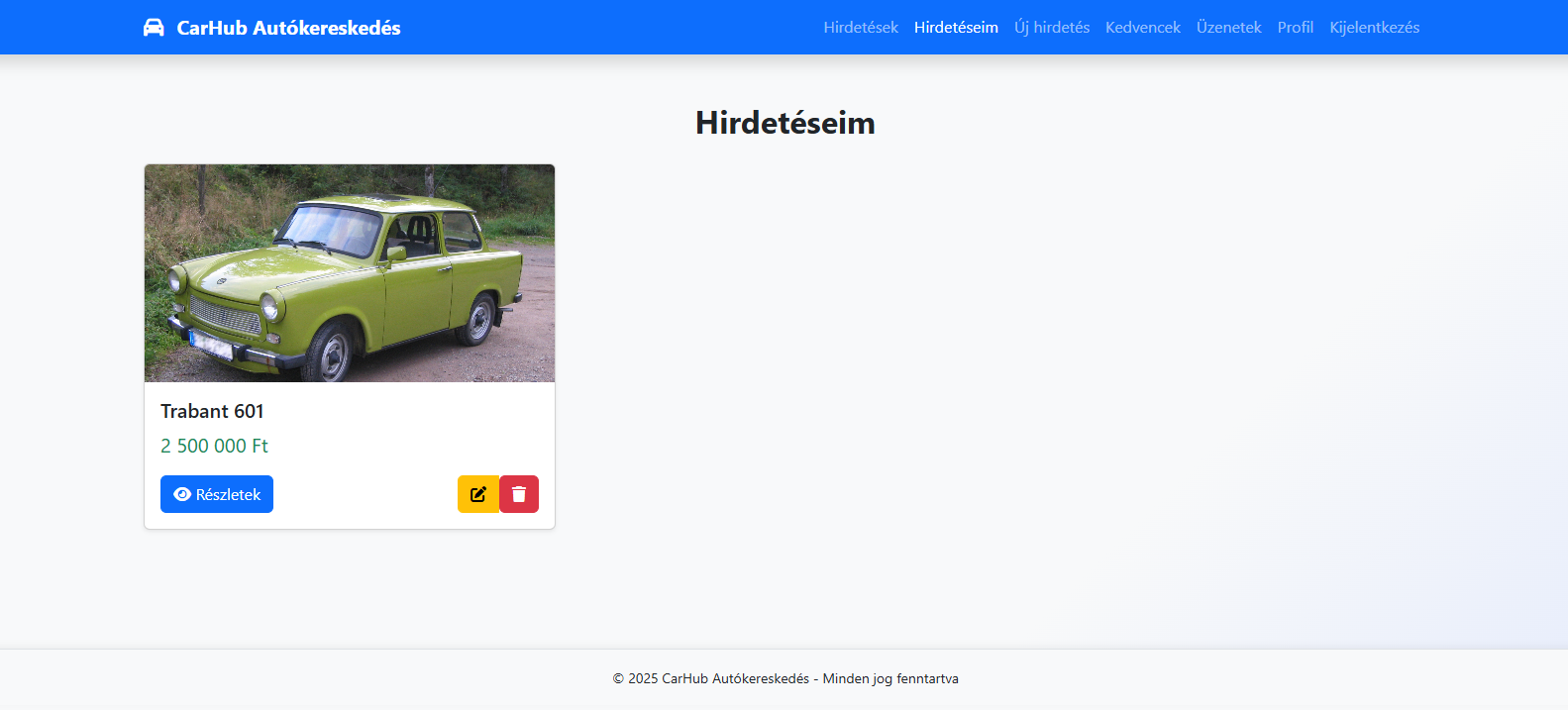
## Saját hirdetések kezelése

- A „Hirdetéseim” menüpontban láthatod az általad feltöltött autókat.

Itt lehetőséged van:

* Szerkeszteni a hirdetést (adatok, képek módosítása)
* Törölni a hirdetést (a képek is törlődnek)

- A hirdetések mellett láthatod a megtekintések számát, az érdeklődők üzeneteit.

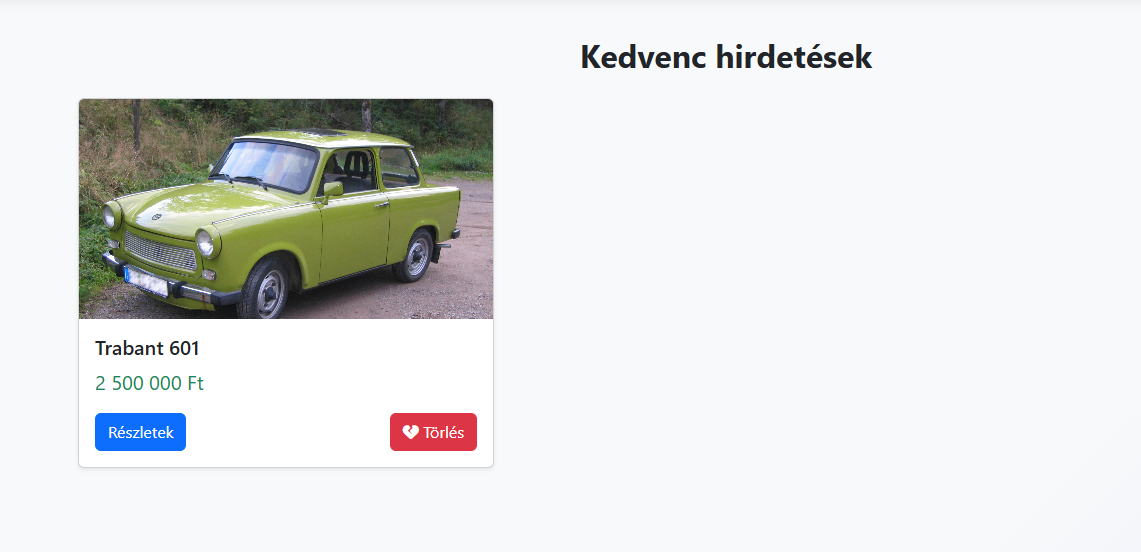


5. ábra Saját hirdetések kezelése

## Kedvencek kezelése

- Bármely hirdetésnél a „Kedvencekhez adás” gombbal elmentheted az autót a kedvenceid közé.

- A „Kedvencek” menüpontban listázva láthatod az összes elmentett hirdetést. Innen egy kattintással eltávolíthatod a kedvencek közül a hirdetést.



6. ábra Kedvenc hirdetések

## Üzenetküldés

- Egy hirdetés adatlapján keresztül közvetlenül üzenetet küldhetsz a hirdetőnek.

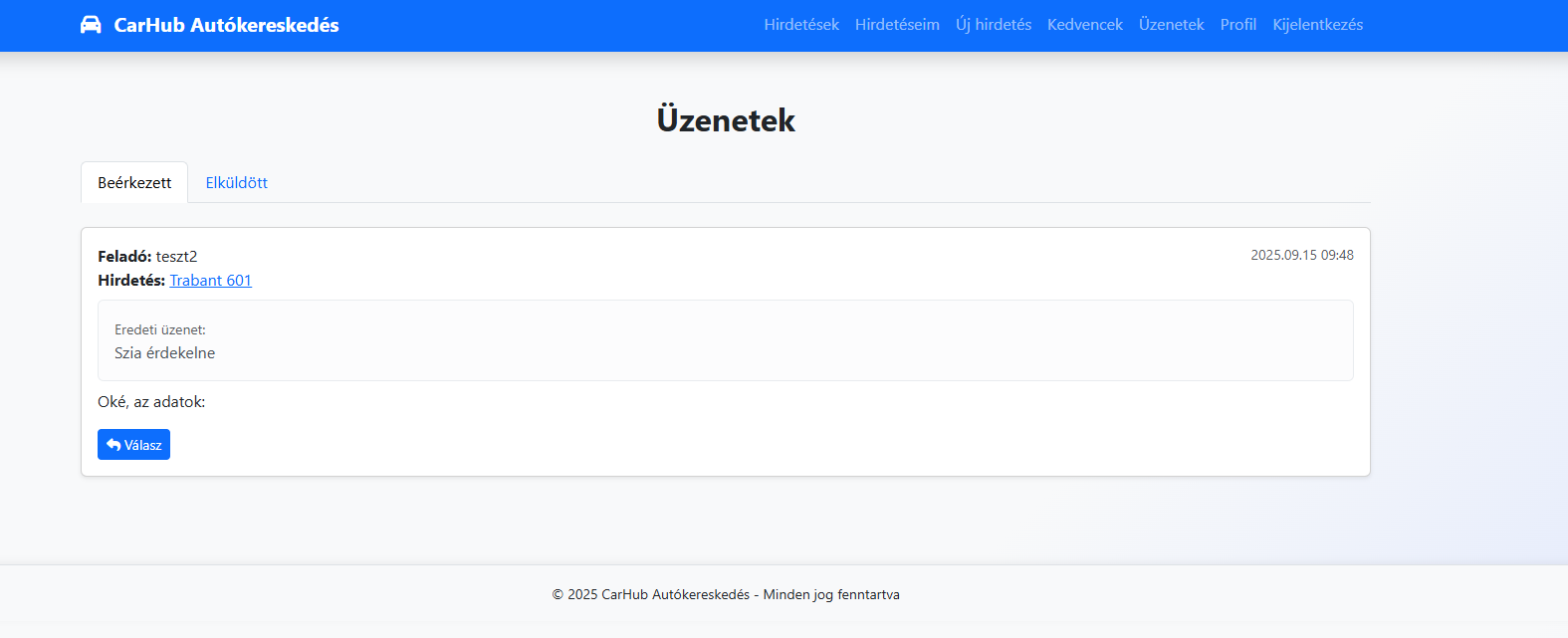
- Az üzenetküldő űrlapon megadhatod a kérdésedet vagy ajánlatodat.

- Az „Üzenetek” menüpontban két fül található:

- Beérkezett üzenetek: Más felhasználóktól kapott üzenetek, válaszolhatsz is rájuk.

- Elküldött üzenetek: Általad küldött üzenetek, követheted a válaszokat.

- Egy üzenetre válaszolva a rendszer automatikusan kezeli a válaszláncot.

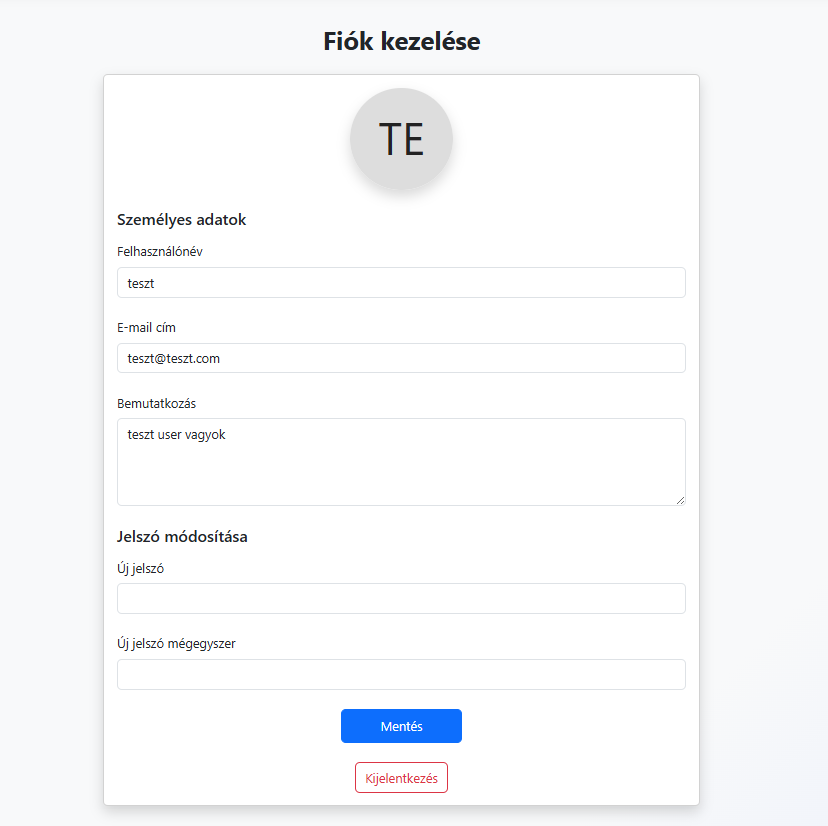


7. ábra Üzenetek

## Profil szerkesztése

- A profiloldalon módosíthatod a bemutatkozásodat, illetve szükség esetén jelszóemlékeztetőt is kérhetsz.

- A profil adataidat csak te láthatod és módosíthatod.



8. ábra Saját profil kezelése

#### Hibakezelés és visszajelzések

- Minden sikeres vagy sikertelen művelet után a rendszer jól látható visszajelzést ad (zöld vagy piros üzenet).

- Hibás adatbevitel esetén a hibás mezők mellett részletes hibaüzenet jelenik meg.

#### Mobil és asztali használat

- Az oldal reszponzív, így mobiltelefonon, tableten és asztali gépen is kényelmesen használható.

- A menük, űrlapok és listák minden eszközön átláthatóak és könnyen kezelhetőek.

A CarHub célja, hogy minden felhasználó gyorsan, biztonságosan és kényelmesen tudjon autót keresni, hirdetni, kapcsolatot teremteni másokkal, és kezelni saját hirdetéseit. Az oldal minden funkciója magyar nyelven, felhasználóbarát módon érhető el.

# Backend – Szerveroldali technológiák

Laravel (PHP keretrendszer):

A projekt alapját a Laravel adja, amely a PHP nyelv egyik legnépszerűbb, nyílt forráskódú keretrendszere. A Laravel támogatja az MVC (Model-View-Controller) architektúrát, így a kód jól strukturált, könnyen karbantartható és bővíthető. A Laravel beépített eszközei – például az Eloquent ORM, a migrációk, a validációs rendszer és az autentikáció – jelentősen megkönnyítik a fejlesztést és növelik a biztonságot.

Eloquent ORM:

Az adatbázis-műveletekhez az Eloquent ORM-et használjuk, amely lehetővé teszi, hogy az adatbázisban tárolt rekordokat objektumként kezeljük, így a kód olvashatóbb és kevésbé hibalehetőséggel terhelt.

RESTful routing:

Az útvonalak (route-ok) kialakítása a REST elveit követi, így minden funkció logikus, jól átlátható URL-en keresztül érhető el. Ez nemcsak a fejlesztők, hanem a felhasználók számára is átláthatóbbá teszi az oldalt.

# Frontend – Felhasználói felület

Blade sablonmotor:

A dinamikus HTML-oldalak generálásához a Laravel Blade sablonmotorját használjuk. Ez lehetővé teszi a nézetek (views) újrahasznosítását, a komponensek egyszerű kezelését, valamint a logika és a megjelenítés szétválasztását.

Bootstrap:

Az oldal reszponzív, modern megjelenését a Bootstrap CSS keretrendszer biztosítja. Ennek köszönhetően a CarHub minden eszközön – asztali gépen, tableten, mobilon – jól használható, a felület letisztult és egységes.

FontAwesome:

Az ikonok megjelenítéséhez a FontAwesome könyvtárat alkalmazzuk, amely széles választékot kínál modern, könnyen felismerhető ikonokból.

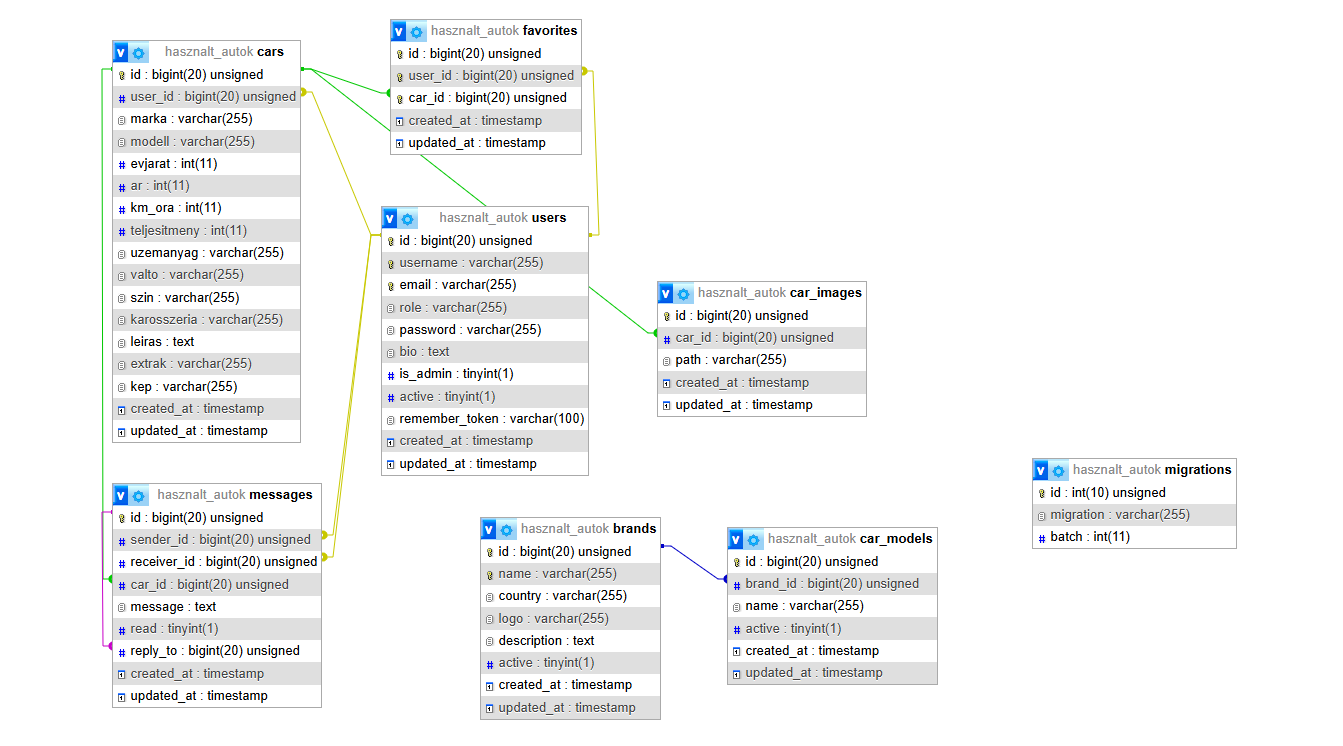
JavaScript (ES6) és AJAX:

Az oldal interaktív elemeihez, például a márka-modell dinamikus választóhoz, JavaScriptet és AJAX technológiát használunk. Így a felhasználói élmény gördülékeny, az oldal részletei frissíthetők anélkül, hogy az egész oldalt újra kellene tölteni.

# Adatbázis és adattárolás

MySQL / SQLite:

A fejlesztés során SQLite adatbázist használunk, de a rendszer könnyen átállítható MySQL-re is. Az adatbázisban tároljuk a felhasználók, autók, márkák, modellek, üzenetek és kedvencek adatait.



9. ábra EER modell az adatbázisról

Fájltárolás (Storage):

A képfeltöltésekhez a Laravel beépített storage rendszerét használjuk, amely a feltöltött képeket a szerver megfelelő könyvtáraiban tárolja, és biztonságos elérést biztosít.

Biztonság és jogosultságkezelés

Laravel Auth:

A felhasználók regisztrációját, bejelentkezését, jogosultságkezelését a Laravel beépített autentikációs rendszere biztosítja. A jelszavak titkosítva tárolódnak, a jogosultságok (pl. hirdetésfeltöltés, szerkesztés) csak bejelentkezett felhasználók számára elérhetők.

Validáció:

Az űrlapok beküldésekor a Laravel validációs rendszere ellenőrzi az adatok helyességét, így elkerülhetők a hibás vagy hiányos adatok, illetve a rosszindulatú próbálkozások.

# Egyéb technológiák

Composer:

A PHP csomagkezelő rendszerével (Composer) kezeljük a külső könyvtárakat és függőségeket, így a projekt könnyen bővíthető és naprakészen tartható.

Git verziókezelés:

A fejlesztés során a Git verziókezelő rendszert használjuk, amely lehetővé teszi a kód változásainak nyomon követését, a csapatmunkát és a biztonsági mentéseket.

Ezeknek a technológiáknak a kombinációja biztosítja, hogy a CarHub Autókereskedés weboldal megbízható, biztonságos, gyors és felhasználóbarát szolgáltatást nyújtson minden látogató számára, miközben a fejlesztők számára is könnyen karbantartható és bővíthető marad.

# Rendszerkövetelmények

A CarHub Autókereskedés weboldal használatához és üzemeltetéséhez az alábbi rendszerkövetelmények szükségesek:

## Felhasználói oldal (böngésző)

A weboldal modern, szabványos böngészőben használható, például:

* Google Chrome
* Mozilla Firefox
* Microsoft Edge
* Safari

További felhasználói követelmények:

* JavaScript: Az oldal számos funkciója JavaScriptet igényel, ezért a böngészőben engedélyezni kell a JavaScript futtatását.
* Cookie-k engedélyezése: A bejelentkezéshez és a felhasználói élményhez szükséges a cookie-k engedélyezése.
* Internetkapcsolat: Az oldal használatához folyamatos internetkapcsolat szükséges.

## Szerveroldal (üzemeltetés)

- Operációs rendszer: Linux, Windows vagy macOS (ajánlott: Linux szerver)

- Webszerver: Apache 2.4+ vagy Nginx

- PHP: PHP 8.1 vagy újabb verzió, a következő bővítményekkel:

- OpenSSL

- PDO

- Mbstring

- Tokenizer

- XML

- Ctype

- JSON

- Fileinfo

A PHP csomagkezelő (Composer) telepítése szükséges a függőségek kezeléséhez.

Tárhely: Legalább 200 MB szabad tárhely a rendszer és a feltöltött képek számára.

- Jogosultságok: A storage és storage könyvtárak írhatóak legyenek a webszerver számára a képfeltöltéshez és cache-eléshez.

## Egyéb

- SSL tanúsítvány: Az oldal biztonságos használatához ajánlott HTTPS-t használni, ehhez érvényes SSL tanúsítvány szükséges.

- E-mail küldés: A jelszóemlékeztető és egyéb értesítések küldéséhez SMTP szerver vagy külső e-mail szolgáltatás (pl. Gmail, SendGrid) beállítása szükséges.

Ezek a követelmények biztosítják, hogy a CarHub weboldal zökkenőmentesen, biztonságosan és gyorsan működjön mind a felhasználók, mind az üzemeltetők számára.

# Adminisztrációs felület

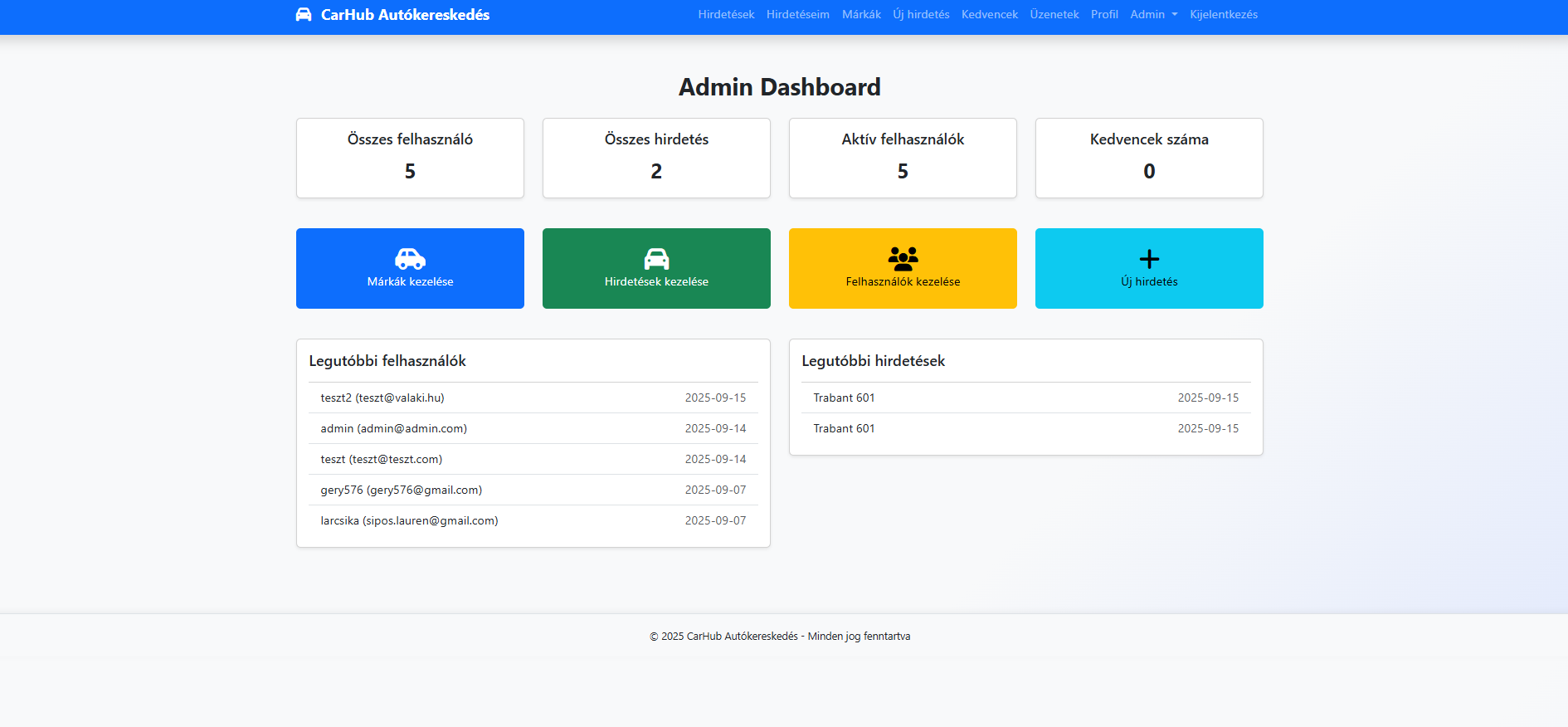
Az adminisztrációs felület a webalkalmazás azon része, amely kizárólag az **adminisztrátorok** számára elérhető. A felület célja, hogy az admin könnyen kezelhesse az oldal tartalmát és felhasználóit.

## Funkciók

1. **Áttekintő dashboard**

Az admin belépés után egy főoldalt lát, ahol kártyákon jelennek meg a legfontosabb statisztikák:

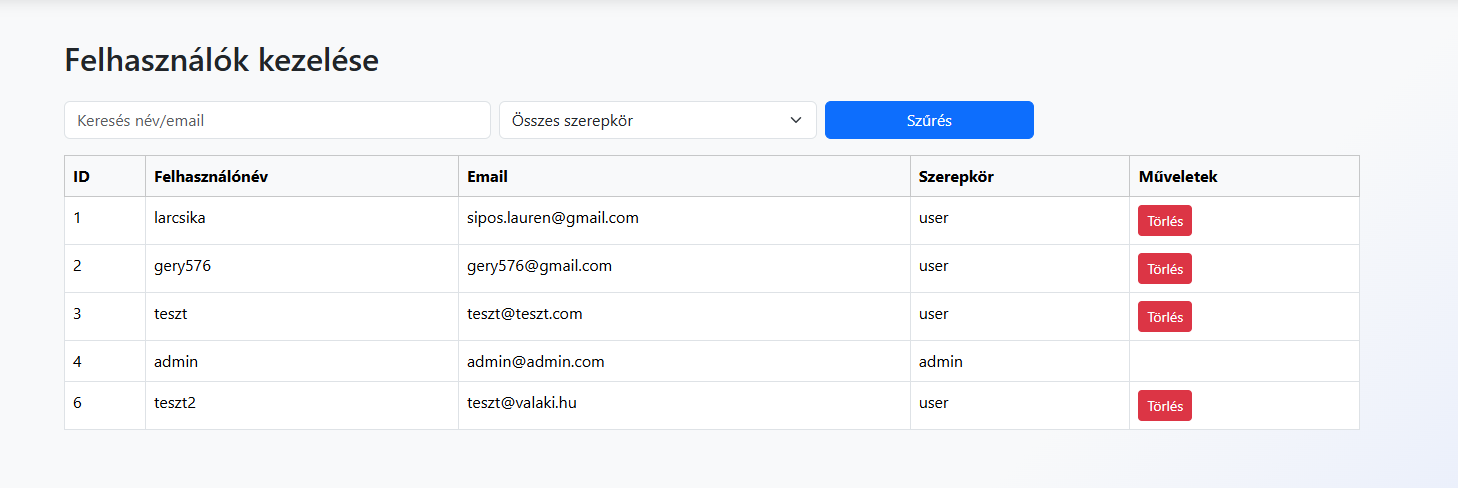
* Felhasználók száma
* Hirdetések száma
* Aktív felhasználók száma
* Kedvencek száma



10. ábra Admin DashBoard

1. **Felhasználók kezelése**

* Listázás, keresés és szűrés (pl. aktív/inaktív felhasználók)
* Felhasználók törlése



11. ábra Felhasználók kezelése

1. **Hirdetések kezelése**

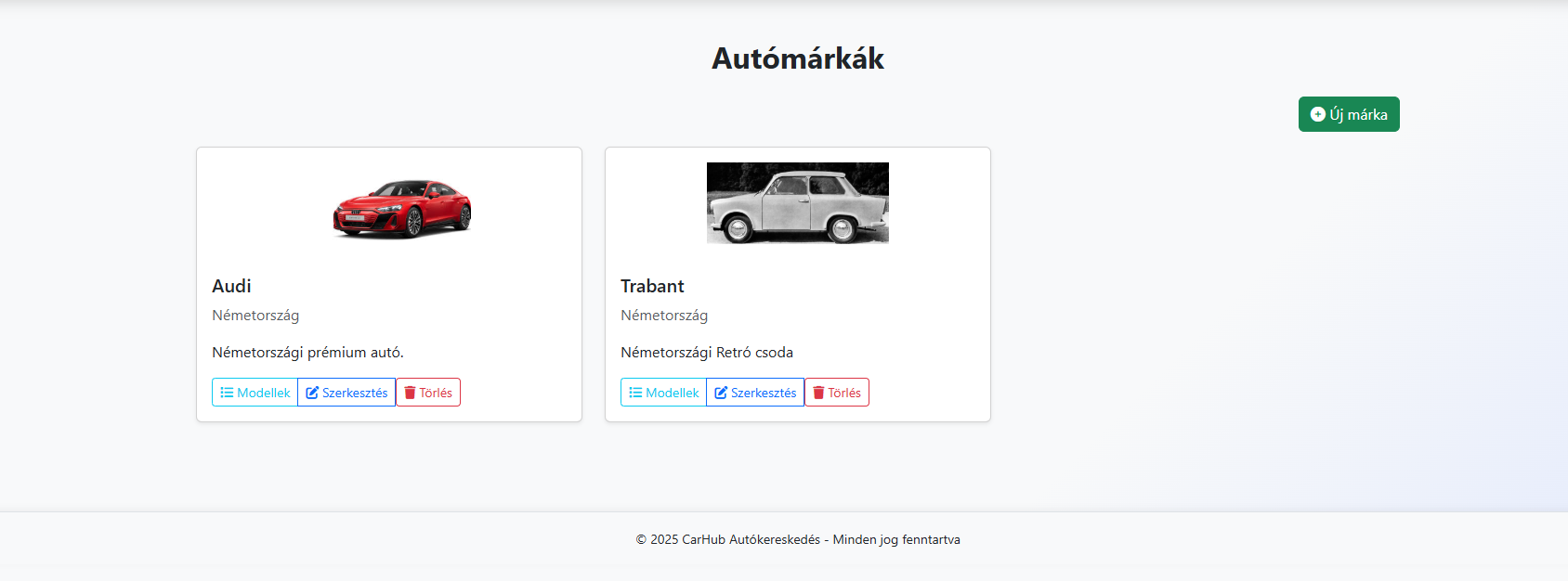
* Az összes autóhirdetés listázása
* Hirdetés törlése, ha szükséges



12. ábra Hirdetések kezelése

1. **Márkák kezelése**

* Márkák listázása
* Új márkák létrehozása
* Márkák törlése



13. ábra Márkák kezelése

## Biztonság és jogosultságok

* Az admin felület csak bejelentkezett, admin szerepkörű felhasználók számára elérhető.
* A jogosultság-ellenőrzést middleware biztosítja, amely minden admin route-nál ellenőrzi a felhasználó szerepkörét.

## Felhasználói élmény

* Egyszerű, áttekinthető kártya-alapú dizájn
* Gyors hozzáférés a legfontosabb funkciókhoz
* Külön felületek a felhasználók, hirdetések és márkák kezeléséhez

# Telepítési útmutató

Az alábbi lépések segítségével telepítheted és elindíthatod a CarHub Autókereskedés webalkalmazást saját gépeden, akár XAMPP-pal is.

## Előfeltételek

- XAMPP (tartalmazza a PHP-t, Apache-ot, MySQL-t)

- Composer (PHP csomagkezelő)

- Git (opcionális, verziókezeléshez)

## Forráskód letöltése

Másold vagy klónozd a projektet a kívánt könyvtárba, például a htdocs mappába:

git clone <repo-url>

vagy másold a teljes projektmappát a htdocs alá.

## Függőségek telepítése

Nyiss egy parancssort a projekt gyökerében, majd futtasd:

composer install

## Környezeti változók beállítása

Másold az .env.example fájlt .env néven:

copy .env.example .env

Állítsd be az adatbázis kapcsolatot az .env fájlban, például:

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=carhub

DB\_USERNAME=root

DB\_PASSWORD=

A DB\_DATABASE értékét állítsd arra az adatbázisra, amit a phpMyAdmin-ban létrehozol.

## Adatbázis létrehozása XAMPP-pal

1. Indítsd el a XAMPP-ot, majd az Apache és MySQL szolgáltatásokat.

2. Nyisd meg a http://localhost/phpmyadmin oldalt.

3. Hozz létre egy új adatbázist (pl. carhub néven, utf8\_general\_ci karakterkészlettel).

## Alkalmazás kulcs generálása

A projekt mappájában futtasd: php artisan key:generate

## Adatbázis migrációk futtatása

Futtasd: php artisan migrate

## Tárhely linkelése a képekhez

php artisan storage:link

## Az alkalmazás futtatása XAMPP-pal

1. Győződj meg róla, hogy a projekt a htdocs mappában van.

2. A böngészőben nyisd meg: http://localhost/autokereskedes/public

3. Ha szeretnéd, hogy a public mappa legyen az alapértelmezett, állítsd be a XAMPP Apache konfigurációjában a DocumentRoot-ot a projekt public mappájára.

## Jogosultságok Windows alatt

A storage és cache mappák írhatóak legyenek (Windows alatt ez általában alapértelmezett).

A fenti lépések elvégzése után a CarHub alkalmazás XAMPP-pal is gond nélkül futtatható.

## Adatbázis struktúra

A CarHub Autókereskedés alkalmazás relációs adatbázist használ, amelyben az adatok táblákban, egymással kapcsolatban tárolódnak. Az alábbiakban bemutatjuk a főbb adatbázis-táblákat és azok kapcsolatait.

## Főbb táblák és mezők

#### 1. users

A felhasználók adatait tárolja.

- id – egyedi azonosító

- username – felhasználónév

- email – e-mail cím

- password – jelszó (titkosítva)

- bio – bemutatkozás

- active – aktív státusz (1/0)

- ...egyéb Laravel által kezelt mezők (pl. timestamps)

#### 2. cars

Az autóhirdetéseket tartalmazza.

- user\_id – a hirdető felhasználó azonosítója (kapcsolat a users táblához)

- marka – márka neve (szöveg)

- modell – modell neve (szöveg)

- evjarat – évjárat

- ar – ár

- leiras – leírás

- kep – fő kép (opcionális, szöveg)

- uzemanyag – üzemanyag típusa

- km\_ora – futott kilométer

- teljesitmeny – teljesítmény

- valto – váltó típusa

- szin – szín

- karosszeria – karosszéria típus

- extrak – extrák (opcionális)

- ...egyéb mezők

#### 3. brands

Az autómárkákat tartalmazza.

- name – márkanév

- country – származási ország

- logo – logó fájlneve (opcionális)

- description – leírás (opcionális)

- active – aktív státusz

#### 4. car\_models

Az egyes márkákhoz tartozó modelleket tartalmazza.

- brand\_id – kapcsolódó márka azonosítója (kapcsolat a brands táblához)

- name – modellnév

#### 5. car\_images

Az autókhoz feltöltött képeket tárolja.

- car\_id – kapcsolódó autó azonosítója (kapcsolat a cars táblához)

- path – kép fájlneve

#### 6. favorites

A felhasználók kedvenc hirdetéseit tartalmazza.

- user\_id – felhasználó azonosítója (kapcsolat a users táblához)

- car\_id – autó azonosítója (kapcsolat a cars táblához)

#### 7. messages

A felhasználók közötti üzenetváltásokat tárolja.

- sender\_id – küldő felhasználó azonosítója (kapcsolat a users táblához)

- receiver\_id – címzett felhasználó azonosítója (kapcsolat a users táblához)

- car\_id – kapcsolódó autó azonosítója (opcionális)

- message – üzenet szövege

- read – olvasottság (opcionális)

- reply\_to – válaszolt üzenet azonosítója (opcionális, kapcsolat önmagára)

## Kapcsolatok

- Egy felhasználónak több autóhirdetése lehet (users –< cars)

- Egy autóhoz több kép tartozhat (cars –< car\_images)

- Egy márkához több modell tartozik (brands –< car\_models)

- Egy felhasználónak több kedvenc hirdetése lehet (users –< favorites >– cars)

- Az üzenetek a felhasználók között jönnek létre, opcionálisan egy autóhoz kapcsolódva, és támogatják a válaszláncokat is.

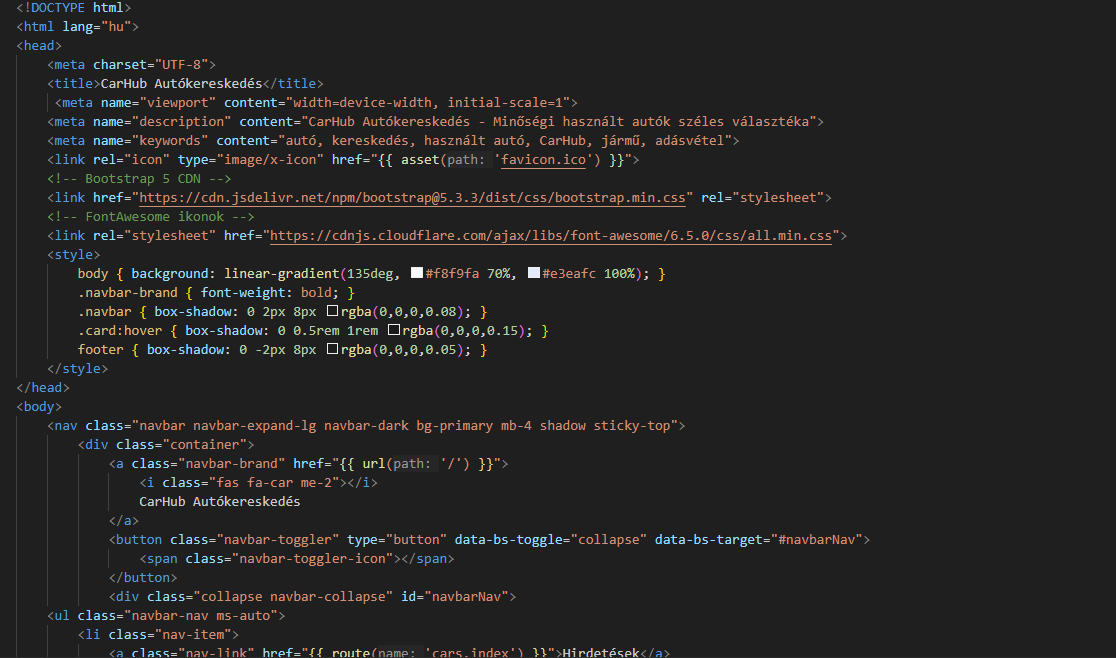
Ez a struktúra biztosítja, hogy az alkalmazás minden funkciója (hirdetés, képkezelés, kedvencek, üzenetek, márkák és modellek) hatékonyan, átláthatóan és biztonságosan működjön.

# Frontend megvalósítás

A CarHub Autókereskedés webalkalmazás felhasználói felülete (frontend) modern, reszponzív és könnyen kezelhető kialakítást kapott, amely a felhasználói élményt helyezi előtérbe. A frontend fejlesztése során a következő főbb technológiákat és megoldásokat alkalmaztuk:

## Blade sablonmotor

A Laravel Blade sablonmotorja biztosítja a dinamikus HTML-oldalak generálását. A sablonok lehetővé teszik az ismétlődő elemek (pl. fejléc, lábléc, navigáció) újrahasznosítását, valamint a logika és a megjelenítés szétválasztását. A nézetek áttekinthetőek, könnyen karbantarthatók és bővíthetők.



14. ábra Layout.blade kód részlet

## Bootstrap

A felület kialakításához a Bootstrap CSS keretrendszert használtuk, amely garantálja a reszponzív, mobilbarát megjelenést. Az űrlapok, gombok, kártyák, táblázatok és egyéb elemek egységes stílust kaptak, így az oldal minden eszközön (asztali gép, tablet, mobil) jól használható.

## Interaktív elemek – JavaScript és AJAX

A felhasználói élmény fokozása érdekében több helyen alkalmaztunk JavaScriptet és AJAX technológiát:

- A hirdetésfeltöltő és szűrő űrlapokon a márka kiválasztása után a modellek listája automatikusan frissül AJAX-kéréssel, így a felhasználónak nem kell újratöltenie az oldalt.

- A kedvencekhez adás, üzenetküldés, képfeltöltés és egyéb műveletek során az oldal gyors visszajelzést ad a felhasználónak.

## Felhasználói nézetek

A frontend különböző nézeteket biztosít a főbb funkciókhoz:

- Hirdetések listázása: Kártyás elrendezés, szűrőkkel és lapozóval.



15. ábra Hirdetések listázása

- Hirdetés részletei: Nagy képek, részletes adatok, eladó elérhetősége, üzenetküldési lehetőség.

- Hirdetés feltöltése/szerkesztése: Többmezős űrlap, képfeltöltés, validációs hibák megjelenítése.

- Kedvencek: Csak a felhasználó által elmentett hirdetések listája.

- Üzenetek: Beérkezett és elküldött üzenetek tabosított nézetben, válasz lehetőséggel.

- Profil: Személyes adatok, saját hirdetések, profil szerkesztése.

## Ikonok és vizuális elemek

A vizuális élményt a FontAwesome ikonok teszik teljessé, amelyek segítik a navigációt és a funkciók gyors felismerését.

## Hibakezelés és visszajelzés

Az űrlapoknál a validációs hibák azonnal, jól láthatóan jelennek meg. Sikeres műveletek után (pl. hirdetés feltöltése, üzenet küldése) a felhasználó visszajelző üzenetet kap.

A frontend megvalósítás célja, hogy a CarHub minden felhasználó számára átlátható, gyors és élvezetes felületet biztosítson, amely támogatja az autóhirdetések böngészését, kezelését és a felhasználók közötti kommunikációt.

# Backend megvalósítás

A CarHub autókereskedés webalkalmazás backendje a Laravel PHP keretrendszerre épül, amely modern, biztonságos és jól strukturált fejlesztési környezetet biztosít. Az alkalmazás szerveroldali logikája az MVC (Model-View-Controller) architektúra szerint épül fel, amely elkülöníti az adatkezelést, az üzleti logikát és a felhasználói felületet.

## Modellek és adatbázis

Az adatbázisban tárolt entitásokat Eloquent modellek reprezentálják, mint például a User, Car, Brand, CarModel, CarImage, Favorite és Message. Ezek a modellek felelősek az adatok lekérdezéséért, mentéséért, kapcsolatok kezeléséért (pl. egy autóhoz több kép tartozhat, egy márkához több modell kapcsolódik). Az adatbázis-migrációk biztosítják a táblák létrehozását és szerkezetük verziókövetését.

## Vezérlők (Controllers)

A vezérlők kezelik a felhasználói kéréseket, validálják az adatokat, és meghívják a megfelelő modelleket vagy nézeteket. Például:

- A CarController felelős az autók listázásáért, szűréséért, feltöltéséért, szerkesztéséért és törléséért.

- A BrandController és CarModelController a márkák és modellek kezelését végzik.

- A FavoriteController a kedvenc hirdetések kezelését biztosítja.

- A MessageController az üzenetküldést és -fogadást valósítja meg.

- A UserController a regisztráció, bejelentkezés, profilkezelés és jelszóemlékeztető funkciókat látja el.

## Validáció és hibakezelés

A backend minden adatbeviteli ponton (pl. regisztráció, autó feltöltése, üzenetküldés) validálja a felhasználói adatokat. A validációs szabályok biztosítják, hogy csak megfelelő formátumú és tartalmú adatok kerüljenek az adatbázisba. Hibás adatok esetén a rendszer visszajelzést ad a felhasználónak, és megőrzi a korábban megadott adatokat.

## Jogosultságkezelés és biztonság

A Laravel beépített authentikációs rendszere gondoskodik a felhasználók bejelentkezéséről, regisztrációjáról, jogosultságainak ellenőrzéséről. Az egyes útvonalak (route-ok) middleware-ekkel védettek, így például autó feltöltése, szerkesztése vagy törlése csak bejelentkezett felhasználók számára elérhető. Az adatok védelme érdekében a jelszavak titkosítva tárolódnak.

## RESTful útvonalak és API-k

Az alkalmazás útvonalai RESTful elvek szerint szerveződnek, így átlátható és könnyen bővíthető a rendszer. Az autómodellek dinamikus betöltéséhez egy egyszerű JSON API is készült (/api/brands/{brand}/models), amely AJAX hívásokkal szolgálja ki a frontend igényeit.

## Fájlkezelés

Az autókhoz és márkákhoz tartozó képek feltöltése és tárolása a Laravel storage rendszerén keresztül történik. A képek a storage mappában érhetők el, és a rendszer automatikusan kezeli a fájlok törlését is, ha egy autó vagy márka törlésre kerül.

## Üzenetküldés és kapcsolatok

A felhasználók egymás között üzeneteket küldhetnek az autóhirdetésekhez kapcsolódóan. Az üzenetekhez tartozó kapcsolatok (küldő, címzett, autó, válaszolt üzenet) Eloquent relációkkal valósulnak meg, így könnyen lekérdezhetők a beérkezett és elküldött üzenetek, valamint a válaszláncok.

## Tesztelés és hibakeresés

A backend működését PHPUnit tesztek segítik, amelyek a tests mappában találhatók. Ezek a tesztek biztosítják, hogy a főbb funkciók (pl. regisztráció, bejelentkezés, autó feltöltése) hibamentesen működjenek, és a fejlesztés során könnyen ellenőrizhető legyen a rendszer stabilitása.

A backend megvalósítás során a fő szempontok a biztonság, a megbízhatóság, a könnyű bővíthetőség és a felhasználóbarát működés voltak. A Laravel keretrendszer és az MVC architektúra lehetővé teszi, hogy a rendszer átlátható, jól karbantartható és hosszú távon is fejleszthető legyen.

# Biztonság

A CarHub autókereskedés webalkalmazás fejlesztése során kiemelt figyelmet kapott a biztonság, hogy a felhasználók adatai és a rendszer működése védett legyen a leggyakoribb támadásokkal és visszaélésekkel szemben.

## Felhasználói hitelesítés és jogosultságkezelés

A Laravel beépített authentikációs rendszere gondoskodik a felhasználók regisztrációjáról, bejelentkezéséről, kijelentkezéséről és a jogosultságok kezeléséről. Az érzékeny funkciók (pl. autó feltöltése, szerkesztése, törlése, üzenetküldés) csak bejelentkezett felhasználók számára érhetők el, amit middleware réteg biztosít.

## Jelszavak védelme

A felhasználói jelszavak soha nem tárolódnak tiszta szövegként az adatbázisban. A Laravel automatikusan bcrypt algoritmussal titkosítja a jelszavakat, így azok visszafejtése gyakorlatilag lehetetlen. Jelszóemlékeztető funkció is elérhető, amely biztonságos módon kezeli a jelszó-visszaállítást.

## Adatvalidáció és hibakezelés

Minden adatbeviteli ponton (regisztráció, bejelentkezés, autó feltöltése, üzenetküldés stb.) szigorú szerveroldali validáció történik. Ez megakadályozza, hogy hibás vagy rosszindulatú adatok kerüljenek a rendszerbe. A validációs hibákról a felhasználó részletes visszajelzést kap.

## CSRF védelem

Az alkalmazás minden űrlapja automatikusan tartalmaz CSRF (Cross-Site Request Forgery) tokent, amelyet a Laravel ellenőriz minden POST, PUT, DELETE kérésnél. Ez megakadályozza, hogy külső weboldalak jogosulatlanul hajtsanak végre műveleteket a felhasználó nevében.

## SQL injection elleni védelem

Az Eloquent ORM és a Laravel query builder automatikusan gondoskodik az SQL lekérdezések paramétereinek megfelelő escape-eléséről, így a rendszer védett az SQL injection támadásokkal szemben.

## Fájlkezelés és jogosultságok

A feltöltött képek és fájlok a storage mappában, elkülönítve tárolódnak. A rendszer csak képfájlokat enged feltölteni, és méretkorlátot is alkalmaz. A fájlokhoz való hozzáférés a Laravel storage rendszerén keresztül történik, így illetéktelenek nem férhetnek hozzá közvetlenül.

## XSS elleni védelem

A Blade sablonmotor alapértelmezetten escape-eli a változók tartalmát, így a felhasználók által megadott adatok nem jelenhetnek meg HTML-kódként, ezzel megelőzve a cross-site scripting (XSS) támadásokat.

## Hozzáférés-szabályozás

A vezérlőkben és az útvonalaknál (route-oknál) middleware-ek gondoskodnak arról, hogy csak a megfelelő jogosultságokkal rendelkező felhasználók érhessenek el bizonyos funkciókat (pl. csak a saját hirdetését szerkesztheti vagy törölheti a felhasználó).

## Naplózás és hibakövetés

A rendszer minden jelentősebb eseményt és hibát naplóz a laravel.log fájlba, így a fejlesztők könnyen visszakövethetik az esetleges problémákat, és gyorsan reagálhatnak a biztonsági incidensekre.

A fenti biztonsági intézkedések együttesen garantálják, hogy a CarHub alkalmazás megbízhatóan és biztonságosan szolgálja ki a felhasználókat, védve adataikat és a rendszer integritását.

# Tesztelés

A CarHub autókereskedés webalkalmazás fejlesztése során a tesztelés kiemelt szerepet kapott annak érdekében, hogy a rendszer megbízhatóan, hibamentesen és biztonságosan működjön. A Laravel keretrendszer beépített tesztelési eszközei lehetővé teszik az automatizált tesztek írását és futtatását, így a fejlesztők gyorsan ellenőrizhetik a főbb funkciók helyes működését.

## Automatizált tesztek

Az alkalmazásban a tests mappában találhatók az automatizált tesztek, amelyek két fő csoportra oszthatók:

- Unit tesztek: Az egyes modellek, függvények, logikai egységek működését ellenőrzik izolált környezetben.

- Feature tesztek: A teljes felhasználói folyamatokat, például a regisztrációt, bejelentkezést, autó feltöltését, üzenetküldést, kedvencek kezelését tesztelik, szimulálva a valós felhasználói interakciókat.

## Tesztelési folyamat

A tesztek futtatása egyszerűen elvégezhető a következő parancs segítségével:

php artisan test

Ez a parancs lefuttatja az összes unit és feature tesztet, és részletes visszajelzést ad a sikeres és sikertelen tesztesetekről.

## Tesztelési területek

A legfontosabb tesztelt funkciók:

- Felhasználói regisztráció, bejelentkezés, jogosultságkezelés

- Autóhirdetés feltöltése, szerkesztése, törlése

- Márkák és modellek kezelése

- Üzenetküldés és üzenetek listázása

- Kedvencek hozzáadása és eltávolítása

- Adatvalidáció, hibakezelés, jogosultságok ellenőrzése

## Manuális tesztelés

Az automatizált tesztek mellett manuális tesztelés is történt, különös tekintettel a felhasználói felületre, a reszponzivitásra, a hibakezelésre és a biztonsági funkciókra. A fejlesztők minden főbb funkciót kipróbáltak különböző böngészőkben és eszközökön.

## Hibakezelés és naplózás

A rendszer minden jelentősebb hibát naplóz, így a fejlesztők gyorsan azonosíthatják és javíthatják az esetlegesen előforduló problémákat. A naplófájlok a laravel.log útvonalon érhetők el.

A tesztelési folyamatnak köszönhetően a CarHub alkalmazás stabil, megbízható és felhasználóbarát, így a felhasználók biztonságosan és gördülékenyen használhatják az oldal minden funkcióját.

# Továbbfejlesztési lehetőségek

A CarHub autókereskedés webalkalmazás jelenlegi formájában egy stabil, jól használható alapot biztosít a használt autók hirdetésére, kezelésére és a felhasználók közötti kommunikációra. Az alábbiakban összegyűjtöttem néhány lehetséges továbbfejlesztési irányt, amelyekkel a rendszer funkcionalitása, felhasználói élménye és piaci értéke tovább növelhető.

#### 1. Képgaléria és képszerkesztés fejlesztése

- Több kép feltöltésének támogatása egy hirdetéshez, képek sorrendjének módosítása.

- Képek vágása, átméretezése, előnézeti kép kiválasztása.

- Drag & drop képfeltöltés, képek törlése szerkesztés közben.

#### 2. Részletesebb kereső- és szűrőrendszer

- Összetett szűrési lehetőségek: ár, évjárat, futott kilométer, üzemanyag, váltó, extrák, szín, karosszéria típus.

- Teljes szöveges keresés a leírásban és extrákban.

- Dinamikus AJAX-alapú szűrés, azonnali eredményfrissítéssel.

#### 3. Felhasználói profilok bővítése

- Profilkép feltöltése, részletesebb bemutatkozás.

- Felhasználói értékelések, visszajelzések (pl. eladók/vevők értékelése).

- Saját hirdetések statisztikái (megtekintések, érdeklődések száma).

#### 4. Üzenetküldés fejlesztése

- Valós idejű (websocketes) chat funkció.

- Üzenetek olvasottságának visszajelzése, értesítések új üzenetekről.

- Mellékletek (pl. képek, dokumentumok) küldése üzenetben.

#### 5. Adminisztrációs felület

- Felhasználók, hirdetések, márkák, modellek, üzenetek kezelése admin oldalon.

- Moderációs eszközök: hirdetések jóváhagyása, tiltása, jelentések kezelése.

- Statisztikák, riportok, forgalmi adatok megjelenítése.

#### 6. Mobilbarát és natív alkalmazás

- Reszponzív design továbbfejlesztése, mobilos felhasználói élmény optimalizálása.

- PWA (Progresszív Webalkalmazás) vagy natív mobilapp (Android/iOS) fejlesztése.

#### 7. E-mail és push értesítések

- Automatikus e-mail értesítések új üzenetekről, hirdetés státusz változásáról.

- Push értesítések támogatása böngészőben és mobilon.

#### 8. Fizetős szolgáltatások, prémium funkciók

- Kiemelt hirdetések, előresorolás, extra megjelenítési lehetőségek.

- Hirdetés kiemelés, statisztikák, hirdetésfigyelő funkciók.

- Online fizetési integráció (pl. Stripe, PayPal).

#### 9. Biztonság további erősítése

- Kétfaktoros hitelesítés (2FA) bevezetése.

- Részletes naplózás, audit trail, gyanús aktivitások figyelése.

- CAPTCHA a regisztrációnál, bejelentkezésnél, űrlapoknál.

#### 10. API és külső integrációk

- Nyilvános REST API fejlesztése mobilapphoz vagy külső partnerekhez.

- Külső autó-adatbázisok, ár-összehasonlító oldalak integrációja.

- Térképes keresés, Google Maps vagy OpenStreetMap integráció.

#### 11. Többnyelvűség, lokalizáció

- Többnyelvű felület (pl. angol, német) támogatása.

- Helyi pénznemek, dátumformátumok, mértékegységek kezelése.

#### 12. Teljesítmény és skálázhatóság

- Gyorsítótárazás (cache), optimalizált adatbázis-lekérdezések.

- Képek CDN-en keresztüli kiszolgálása.

- Nagyobb forgalom esetén horizontális skálázás, load balancer alkalmazása.

Ezek a továbbfejlesztési lehetőségek segítenek abban, hogy a CarHub alkalmazás a jövőben még versenyképesebb, felhasználóbarátabb és üzletileg sikeresebb legyen. Az egyes fejlesztési irányok igény szerint, modulárisan is bevezethetők, így a rendszer könnyen bővíthető és testreszabható marad.

# Függelék

Ez a függelék összefoglalja a rendszer főbb technikai részleteit, segíti a fejlesztők és üzemeltetők munkáját, valamint támogatja a további bővítéseket és karbantartást.

## Főbb modellek és kapcsolataik

- User: felhasználók adatai, kapcsolódik autókhoz (cars), kedvencekhez (favorites), üzenetekhez (sentMessages, receivedMessages).

- Car: autóhirdetések, kapcsolódik felhasználóhoz (user), képekhez (images), márkához (név szerint), modellekhez (név szerint).

- Brand: autómárkák, kapcsolódik modellekhez (models), autókhoz (cars).

- CarModel: autómodellek, kapcsolódik márkához (brand).

- CarImage: autóképek, kapcsolódik autóhoz (car).

- Favorite: kedvenc hirdetések, kapcsolódik felhasználóhoz (user) és autóhoz (car).

- Message: üzenetek, kapcsolódik küldőhöz (sender), címzetthez (receiver), autóhoz (car), válaszolt üzenethez (originalMessage).

## Fontosabb útvonalak (routes)

- Publikus:

- /cars – hirdetések listázása

- /cars/{car} – hirdetés részletei

- /user/{user}/cars – adott felhasználó hirdetései

- /api/brands/{brand}/models – márkához tartozó modellek (AJAX)

- /reg, /login, /logout – regisztráció, bejelentkezés, kijelentkezés

- Bejelentkezéshez kötött:

- /feltoltes – új autó feltöltése

- /cars/{car}/edit – hirdetés szerkesztése

- /cars/{car} (PUT/DELETE) – hirdetés módosítása, törlése

- /profile (PUT) – profil szerkesztése

## Validációs szabályok példák

- Márka létrehozás: név kötelező, egyedi; ország kötelező; logó opcionális, max. 2MB, csak kép.

- Autó feltöltés: kötelező mezők (márka, modell, évjárat, ár, stb.), képek csak képformátumban.

- Üzenetküldés: üzenet minimum 10 karakter.

## Hibakezelés és visszajelzések

- Minden űrlap validációs hibáit magyar nyelven jeleníti meg.

- Sikeres műveletek után (pl. hirdetés feltöltése, márka hozzáadása) zöld visszajelző üzenet jelenik meg.

- Hibák naplózása: laravel.log.

## Használt csomagok, technológiák

- Laravel – PHP keretrendszer

- Blade – sablonmotor

- Bootstrap 5 – reszponzív frontend

- FontAwesome – ikonok

- Eloquent ORM – adatbázis-kezelés

- AJAX (fetch API) – dinamikus modellek betöltése

- PHPUnit – tesztelés

- SQLite/MySQL – adatbázis

- XAMPP – fejlesztői környezet

## Példa adatbázis-struktúra (részlet)

- users: id, username, email, password, bio, active, ...

- cars: id, user\_id, marka, modell, evjarat, ar, leiras, ...

- brands: id, name, country, logo, description, active

- car\_models: id, brand\_id, name, active

- car\_images: id, car\_id, path

- favorites: id, user\_id, car\_id

- messages: id, sender\_id, receiver\_id, car\_id, message, read, reply\_to

## Tesztelés

- Alapvető funkciók tesztelése PHPUnit-tel: regisztráció, bejelentkezés, autó feltöltése, üzenetküldés.

- Manuális tesztelés: űrlapok, jogosultságok, képfeltöltés, AJAX működés.

## Hasznos parancsok fejlesztéshez

- php artisan migrate – adatbázis migrációk futtatása

- php artisan db:seed – tesztadatok feltöltése

- php artisan storage:link – storage elérhetővé tétele

- php artisan serve – fejlesztői szerver indítása

## Elérhetőségek, támogatás

- Laravel dokumentáció: https://laravel.com/docs

- Bootstrap dokumentáció: https://getbootstrap.com

- Hibák, kérdések esetén: fejlesztői e-mail vagy GitHub issue

# Ábrajegyzék

[1. ábra Regisztráció 7](#_Toc208838170)

[2. ábra Bejelentkezés 8](#_Toc208838171)

[3. ábra Hirdetések megjelenítése 9](#_Toc208838172)

[4. ábra Új hirdetés feladása 9](#_Toc208838173)

[5. ábra Saját hirdetések kezelése 10](#_Toc208838174)

[6. ábra Kedvenc hirdetések 10](#_Toc208838175)

[7. ábra Üzenetek 11](#_Toc208838176)

[8. ábra Saját profil kezelése 11](#_Toc208838177)

[9. ábra EER modell az adatbázisról 13](#_Toc208838178)

[10. ábra Admin DashBoard 16](#_Toc208838179)

[11. ábra Felhasználók kezelése 17](#_Toc208838180)

[12. ábra Hirdetések kezelése 17](#_Toc208838181)

[13. ábra Márkák kezelése 18](#_Toc208838182)

[14. ábra Layout.blade kód részlet 23](#_Toc208838183)

[15. ábra Hirdetések listázása 24](#_Toc208838184)