

R&J Autókölcsönző - Projekt Dokumentáció

Verzió: 1.0

Dátum: 2025-04-09 (a SQL dump alapján)

Készítők:

- Felhasználói Felület: Jurcsó János Árpád
 - Adminisztrációs Felület: Dávid Roland
-

Tartalomjegyzék

1. Bevezetés
 - 1.1. A Projekt Célja
 - 1.2. A Rendszer Áttekintése
 - 1.3. Célközönség
2. Rendszerarchitektúra és Technológiák
 - 2.1. Architektúra
 - 2.2. Technológiai Stack
3. Funkcionális Követelmények és Megvalósítás
 - 3.1. Felhasználói Funkciók
 - 3.1.1. Regisztráció (register.php, index.php modal)
 - 3.1.2. Bejelentkezés / Kijelentkezés (login.php, logout.php, index.php modal)
 - 3.1.3. Főoldal (index.php)
 - 3.1.4. Járművek Böngészése és Szűrése (jarmuvek.php)
 - 3.1.5. Jármű Részletek (auto_adatok.php)
 - 3.1.6. Jármű Bérlet (jarmuvek.php modal, phpmailer)
 - 3.1.7. Profil Kezelés (profilom.php, modosit_profil.php)
 - 3.1.8. Véleményezés (auto_adatok.php, cseveges.php, kapcsolat.php)
 - 3.1.9. Kapcsolatfelvétel (kapcsolat.php)
 - 3.1.10. Fórum/Gépjárművek (forum.php)
 - 3.1.11. Hűségpontok (husegpontok.php)
 - 3.1.12. Modális Ablakok (modal.php, index.js)
 - 3.2. Adminisztrációs Funkciók
 - 3.2.1. Autentikáció és Hozzáférés
 - 3.2.2. Vezérlőpult Menü (közös elem)
 - 3.2.3. Járműkezelés (autok_kezeles.php, autok_kezeles_modositas.php)
 - 3.2.4. Jogosultságkezelés (admin_jogosultsag.php, admin_jog_mod.php)
 - 3.2.5. Bérletkezelés (admin_berlesek.php, admin_berlesek_mod.php)
 - 3.2.6. Véleménykezelés (admin_velemenyek.php)
 - 3.2.7. Akciókezelés (admin_akciok.php)
 - 3.3. API Végpontok (Profil és Bérletek)
 - 3.3.1. Felhasználói Adatok (api/user.php)
 - 3.3.2. Bérleti Adatok (api/rentals.php)
4. Adatbázis Tervezés
 - 4.1. Adatbázis Séma (autoberles.sql)
 - 4.2. Táblák Leírása

- 4.2.1. felhasználó
 - 4.2.2. járművek
 - 4.2.3. felhasználás
 - 4.2.4. szervizek
 - 4.2.5. berlek
 - 4.2.6. vélemények
 - 4.2.7. akciók
 - 4.3. Kapcsolatok
 - 5. Fájlstruktúra
 - 6. Felhasználói Felület (UI/UX)
 - 6.1. Stíluslapok (style.css, admin.css, stb.)
 - 6.2. Reszponzivitás
 - 6.3. Kliensoldali Interakciók (index.js, járművek.js)
 - 7. Biztonság
 - 7.1. Autentikáció és Autorizáció
 - 7.2. Adatbázis Biztonság (SQL Injection)
 - 7.3. Adatkezelés (Input Sanitization, XSS)
 - 7.4. Jelszókezelés
 - 7.5. Fájlfeltöltés Biztonsága
 - 7.6. További Megfontolások
 - 8. Telepítés és Konfiguráció
 - 9. Tesztelés (test_full_application.py)
 - 10. Jövőbeli Fejlesztési Lehetőségek
 - 11. Összegzés
-

1. Bevezetés

1.1. A Projekt Célja

A projekt célja egy webes alapú autókölcsönző alkalmazás, az "R&J Autókölcsönző" létrehozása volt. Az alkalmazás lehetővé teszi a regisztrált felhasználók számára, hogy megtekintsék a bérelhető járművek kínálatát, szűrjenek a járművek között, részletes adatokat kapjanak egy-egy autóról, véleményeket olvassanak és írjanak, valamint online lefoglalják a kiválasztott járművet a megadott időszakra. Az alkalmazás tartalmaz egy adminisztrációs felületet is, ahol a cég munkatársai kezelhetik a járműflottát, a felhasználókat, a bérleket, a véleményeket és az akciókat.

1.2. A Rendszer Áttekintése

Az R&J Autókölcsönző egy PHP nyelven íródott, MySQL (vagy MariaDB) adatbázist használó webalkalmazás. Két fő részből áll:

- **Felhasználói Felület:** Nyilvánosan elérhető (bizonyos funkciók bejelentkezéshez kötöttek), ahol a látogatók és regisztrált felhasználók böngészhetnek,

regisztrálhatnak, bejelentkezhetnek, bérelhetnek és kezelhetik profiljukat. Ezt a részt Jurcsó János Árpád fejlesztette.

- **Adminisztrációs Felület:** Csak adminisztrátori jogosultsággal rendelkező felhasználók számára elérhető, a rendszer tartalmának és működésének menedzselésére szolgál. Ezt a részt Dávid Roland fejlesztette.

A rendszer alapvető funkciói közé tartozik a felhasználókezelés (regisztráció, bejelentkezés, profil módosítás), a járműkatalógus megjelenítése és szűrése, a bérleti folyamat kezelése (beleértve a dátumütközések ellenőrzését és az e-mailes visszaigazolást), a felhasználói vélemények gyűjtése és megjelenítése, valamint a hűségpontrendszer alapjainak kezelése. Az adminisztrációs felület teljes körű CRUD (Create, Read, Update, Delete) műveleteket biztosít a főbb entitások (járművek, felhasználók, bérletek, vélemények, akciók) felett.

1.3. Célközönség

- **Felhasználók:** Magánszemélyek vagy cégek, akik gépjárművet szeretnének bérelni rövidebb vagy hosszabb távra.
- **Adminisztrátorok:** Az R&J Autó kölcsönző munkatársai, akik felelősek a weboldal tartalmának frissítéséért, a foglalások kezeléséért és a rendszer általános felügyeletéért.

2. Rendszerarchitektúra és Technológiák

2.1. Architektúra

A rendszer egy klasszikus, szerveroldali renderelésen alapuló webalkalmazás architektúrát követ.

- **Prezentációs Réteg:** HTML, CSS (Bootstrap keretrendszerrel kiegészítve) és JavaScript (interaktivitáshoz, pl. modális ablakok, űrlap validáció). A PHP generálja a dinamikus HTML tartalmat.
- **Üzleti Logikai Réteg:** PHP szkriptek kezelik a felhasználói kéréseket, validálják az adatokat, végrehajtják a szükséges műveleteket (pl. bérlet rögzítése, hűségpont számítás) és kommunikálnak az adatréteggel. Különálló PHP fájlok felelnek az egyes funkciókért (pl. `jarmuvek.php`, `login.php`, `admin_autok_kezeles.php`). Az `api/` mappa szkriptjei JSON választ adnak a profil oldal dinamikus betöltéséhez.
- **Adatréteg:** MySQL/MariaDB adatbázis tárolja az alkalmazás adatait (felhasználók, járművek, bérletek stb.). A PHP szkriptek MySQLi kiterjesztéssel (prepared statementek használatával) kommunikálnak az adatbázissal (`db_connection.php`, `adatLekeres.php`).

2.2. Technológiai Stack

- **Szerveroldali nyelv:** PHP (valószínűleg 7.4 vagy újabb, a `password_hash`, `fetch_all`, `??` operátor és a `api/` fájlok JSON kezelése alapján)
- **Adatbázis:** MySQL / MariaDB
- **Webszerver:** Apache vagy Nginx (nem specifikált, de tipikus PHP környezet)

- **Kliensoldali technológiák:**
 - **HTML5**
 - **CSS3** (beleértve a style.css, admin.css, stb. fájlokat)
 - **Bootstrap 5** (a CSS és JS hivatkozások alapján)
 - **JavaScript** (Vanilla JS a index.js, jarmuvek.js fájlokban és inline scriptekben)
- **Szerveroldali könyvtárak:**
 - **PHPMailer** (e-mail küldéshez a bérletnél és admin válaszoknál)

3. Funkcionális Követelmények és Megvalósítás

3.1. Felhasználói Funkciók

- **3.1.1. Regisztráció** (register.php, index.php modal)
 - **Leírás:** Új felhasználók létrehozása. Az index.php-n lévő "Regisztráció" gomb megnyit egy modális ablakot (registerModal), ahol a felhasználó megadhatja adatait.
 - **Megvalósítás:** A register.php fogadja a POST kérést. Ellenőrzi, hogy a jelszavak egyeznek-e, és hogy a felhasználónév nem foglalt-e már. A jelszót a password_hash() függvénnyel hasheli (PASSWORD_DEFAULT algoritmussal). Sikeres validáció után beszúrja az új felhasználót a felhasznalo táblába. Visszajelzést ad a felhasználónak session üzeneteken keresztül (\$_SESSION['alert_message'], \$_SESSION['alert_type']), majd visszairányít az index.php-re.
 - **Adatbázis:** felhasznalo tábla (INSERT).
- **3.1.2. Bejelentkezés / Kijelentkezés** (login.php, logout.php, index.php modal)
 - **Leírás:** Meglévő felhasználók autentikálása és session kezelés. Az index.php-n lévő "Bejelentkezés" gomb megnyitja a loginModal-t. Kijelentkezés a "Kijelentkezés" menüponttal.
 - **Megvalósítás:**
 - **Bejelentkezés:** Az index.php (vagy a dedikált login.php, bár a kód alapján az index.php kezeli a POST-ot) fogadja a POST kérést. Lekérdezi a felhasználót név alapján. A password_verify() függvénnyel ellenőrzi a megadott jelszót a hashelt adatbázisbeli jelszóval. Sikeres azonosítás esetén beállítja a \$_SESSION['felhasznalo_nev'] és \$_SESSION['admin'] változókat. Hibás adatok esetén session üzenetet állít be. Visszairányítás index.php-re.
 - **Kijelentkezés:** A logout.php törli az összes session változót (session_unset(), session_destroy()), majd visszairányít az index.php-re.

- Adatbázis: felhasználó tábla (SELECT).
- 3.1.3. Főoldal (index.php)
 - Leírás: Az alkalmazás kezdőoldala. Tartalmazza a főmenüt, egy üdvözlő üzenetet és egy űrlapot a bérleti időszak megadására, amely a jarmuvek.php-re irányít. Itt találhatóak a bejelentkezési és regisztrációs modális ablakok indítógombjai is. Kezeli a bejelentkezési POST kérést is.
 - Megvalósítás: Session alapján dönti el, hogy a "Bejelentkezés/Regisztráció" vagy a "Profilom/Kijelentkezés" menüpontok jelenjenek meg. Tartalmazza a modális ablakok HTML struktúráját és a modal.php include-ot a session üzenetek megjelenítésére. A dátumválasztó űrlap GET kéréssel küldi tovább az adatokat a jarmuvek.php-re.
- 3.1.4. Járművek Böngészése és Szűrése (jarmuvek.php)
 - Leírás: Listázza a bérelhető járműveket. Lehetőséget biztosít szűrésre átvételi/leadási dátum, járműkategória és ár alapján. A szűrési feltételek a GET paraméterekből érkeznek. Csak bejelentkezett felhasználók érhetik el.
 - Megvalósítás: Ellenőrzi a felhasználói session-t. Feldolgozza a GET paramétereket. Dinamikus SQL lekérdezést épít fel (prepared statementekkel) a jarmuvek, akciók, felhasznalas, és berlesek táblák alapján. Kizárja azokat a járműveket, amelyek a megadott időintervallumban már foglaltak. Megjeleníti az elérhető járműveket kártya nézetben, figyelembe véve az esetleges akciókat (áthúzott ár, akciós ár). Tartalmaz egy lenyitható szűrő űrlapot. Minden kártyán van egy "Bérlés" gomb, ami megnyit egy modális ablakot.
 - Adatbázis: jarmuvek, akciók, felhasznalas, berlesek, felhasználó (SELECT).
- 3.1.5. Jármű Részletek (auto_adatok.php)
 - Leírás: Egy kiválasztott jármű részletes adatlapja. Megjeleníti a jármű adatait (gyártó, típus, motor, évjárat, leírás, ár), a hozzá tartozó képeket galériában (nagyítható, lapozható), valamint a járműhöz írt felhasználói véleményeket. Lehetőséget ad új vélemény írására.
 - Megvalósítás: A GET id paraméter alapján lekérdezi a jármű adatait a jarmuvek táblából. A kep_url JSON mezőt dekódolja és megjeleníti a képeket (első kép kiemelve, többi a modális galériában). Lekérdezi és listázza a kapcsolódó véleményeket a velemenyek táblából. Tartalmaz egy űrlapot új vélemény küldésére, ami a cseveges.php-re küldi az adatokat. Használ JavaScriptet a galéria működtetéséhez és a karakterlimit számláléhoz.
 - Adatbázis: jarmuvek, velemenyek (SELECT).
- 3.1.6. Jármű Bérlés (jarmuvek.php modal, phpmailer)
 - Leírás: A jármű listában vagy a főoldali űrlapon keresztül kezdeményezett bérleti folyamat véglegesítése. A jarmuvek.php-n lévő "Bérlés" gomb megnyit egy modális ablakot (modal).

- **Megvalósítás:** A `jarmuvek.php` kezeli a bérleti űrlapról érkező POST kérést. Az űrlap a modális ablakban jelenik meg, előre kitöltve a felhasználó nevével és email címével (session alapján), valamint az átvételi/leadási dátumokkal (ha voltak szűrési feltételek). A felhasználó megadja a telefonszámát és választ fizetési módot (azonnal/helyszínen - ez a kifizetve mezőt állítja be 0-ra vagy 1-re). A szkript beszűri a bérleti adatokat a berlek tablejéba. Kiszámolja a bérlet teljes költségét (figyelembe véve az akciókat) és a kapott hűségpontokat (költség 10%-a, lefelé kerekítve). Frissíti a felhasználó hűségpontjait a felhasználó tablejében. PHPMailer segítségével visszaigazoló e-mailt küld a felhasználónak a bérlet részleteiről. Visszajelzést ad session üzenetben.
- **Adatbázis:** berlek (INSERT), felhasználó (UPDATE - hűségpont), jarmuvek, akciók (SELECT - ár és kedvezmény lekérdezéshez).
- **Külső szolgáltatás:** SMTP szerver (pl. Gmail) az e-mail küldéshez PHPMailer-en keresztül.
- **3.1.7. Profil Kezelés (`profilom.php`, `modosit_profil.php`)**
 - **Leírás:** Bejelentkezett felhasználók megtekinthetik és módosíthatják saját adataikat, valamint megtekinthetik korábbi bérleteiket és hűségpontjaikat. Adminok számára itt jelenik meg a "Vezérlőpult" gomb.
 - **Megvalósítás:**
 - `profilom.php`: JavaScript fetch kéréseket indít az `api/user.php` és `api/rentals.php` végpontok felé. A kapott JSON adatok alapján dinamikusan tölti fel a profil adatokat (név, email, cím, jogosítvány dátuma, hűségpontok) és a korábbi bérletek listáját egy táblázatba. Ellenőrzi az admin státuszt és megjeleníti az admin gombot, ha szükséges.
 - `modosit_profil.php`: Egy űrlapot jelenít meg a felhasználói adatok módosításához. JavaScript fetch-csel lekéri a jelenlegi adatokat az `api/user.php`-ről és betölti az űrlapba. Az űrlap elküldésekor (submit) JavaScript fetch-csel POST kérést küld az `api/user.php`-nak a FormData objektummal. Lehetőséget ad új jelszó megadására is (a jelszóváltás az `api/user.php`-ban történik). A válasz alapján kezeli a siker/hiba üzeneteket (sessionön keresztül vagy közvetlen visszajelzéssel - a kód alapján session-t próbál beállítani egy `modal.php` hívásnak tűnő fetch-csel, ami nem standard megoldás).
 - **API Hívások:** `api/user.php` (GET, POST), `api/rentals.php` (GET).
- **3.1.8. Véleményezés (`auto_adatok.php`, `cseveges.php`, `kapcsolat.php`)**
 - **Leírás:** Felhasználók véleményt írhatnak konkrét járművekről vagy általános visszajelzést adhatnak a kapcsolat oldalon.
 - **Megvalósítás:**

- **auto_adatok.php:** Tartalmaz egy űrlapot, amely POST kérést küld a cseveges.php-nek. A jarmu_id rejtett mezőben kerül átadásra. Karakterlimit ellenőrzés JavaScripttel.
- **cseveges.php:** Fogadja a POST kérést az auto_adatok.php-ról. Beszúrja a véleményt a velemenyek táblába a megfelelő jarmu_id-val, a felhasználó nevével (sessionből) és az üzenettel. Sikeres mentés után visszairányít az auto_adatok.php oldalra.
- **kapcsolat.php:** Hasonló űrlapot tartalmaz, de ez a vélemény általános (implikált jarmu_id = 0). A kapcsolat.php maga dolgozza fel a POST kérést, beszúrja a velemenyek táblába, session üzenetet állít be, és újratölti az oldalt.
- **Adatbázis: velemenyek (INSERT).**
- **3.1.9. Kapcsolatfelvétel (kapcsolat.php)**
 - **Leírás:** Elérhetőségi információk megjelenítése és egy kapcsolatfelvételi űrlap biztosítása. Csak bejelentkezett felhasználók érhetik el.
 - **Megvalósítás:** Megjeleníti a cég címét, telefonszámát, email címét. Tartalmaz egy űrlapot, ahol a név (sessionből) és email (manuálisan beírandó) mellett üzenetet lehet küldeni. Az űrlap POST kérését a kapcsolat.php saját maga dolgozza fel, az üzenetet a velemenyek táblába menti (implikált jarmu_id = 0), majd session üzenettel jelzi a sikert/hibát és újratölti az oldalt.
 - **Adatbázis: velemenyek (INSERT).**
- **3.1.10. Fórum/Gépjárművek (forum.php)**
 - **Leírás:** Listázza az összes adatbázisban szereplő járművet kártya nézetben, képpel és névvel. A kártyákra kattintva a felhasználó az auto_adatok.php oldalra jut az adott járműhöz.
 - **Megvalósítás:** Lekérdezi az összes járművet a jarmuvek táblából. Ciklussal végigmegy az eredményeken, dekódolja a kep_url JSON-t, és megjeleníti az első képet a kártyán. A kártya egy link, amely az auto_adatok.php-re mutat a megfelelő jarmu_id GET paraméterrel.
 - **Adatbázis: jarmuvek (SELECT).**
- **3.1.11. Hűségpontok (husegpontok.php)**
 - **Leírás:** Tájékoztató oldal a hűségpontrendszer működéséről. Csak bejelentkezett felhasználók érhetik el.
 - **Megvalósítás:** Statikus HTML tartalom, amely leírja a pontgyűjtés és -felhasználás módját. Ellenőrzi a felhasználói sessiont.
- **3.1.12. Modális Ablakok (modal.php, index.js)**
 - **Leírás:** Figyelmeztető és tájékoztató üzenetek, valamint bejelentkezési/regisztrációs űrlapok megjelenítése felugró ablakokban.

- **Megvalósítás:** A modal.php egy PHP include, amely ellenőrzi a `$_SESSION['alert_message']` és `$_SESSION['alert_type']` változókat. Ha léteznek, generál egy alertModal nevű modális ablakot a megfelelő tartalommal és stílussal (success/warning), majd törli a session változókat. JavaScript (index.js openModal, closeModal és inline scriptek) vezérli a modális ablakok megjelenítését és bezárását, beleértve az overlay (háttér elsötétítés) kezelését is. Az index.php és jarmuvek.php is használja ezt a mechanizmust.

3.2. Adminisztrációs Funkciók

• 3.2.1. Autentikáció és Hozzáférés

- **Leírás:** Csak adminisztrátori jogosultsággal rendelkező felhasználók érhetik el az admin felületet.
- **Megvalósítás:** Minden admin oldalon (admin/php/*.php) szerepel egy session ellenőrzés (`session_start(); if ($_SESSION['admin'] == false) { ... header("Location: ..."); exit(); }`). A bejelentkezéskor (index.php vagy login.php) beállított `$_SESSION['admin']` (értéke 1, ha admin) alapján történik az ellenőrzés.

• 3.2.2. Vezérlőpult Menü (közös elem)

- **Leírás:** Az admin oldalak tetején található navigációs menü, amely gyors hozzáférést biztosít a különböző adminisztrációs modulokhoz (Járművek, Jogosultságok, Bérlek, Vélemények, Akciók).
- **Megvalósítás:** Minden admin oldalon szereplő HTML blokk, amely linkeket tartalmaz a többi admin oldalra (autok_kezeles.php, admin_jogosultsag.php, stb.).

• 3.2.3. Járműkezelés (autok_kezeles.php, autok_kezeles_modositas.php)

- **Leírás:** Járművek hozzáadása, listázása, módosítása és törlése.
- **Megvalósítás:**
 - **autok_kezeles.php:**
 - **Hozzáadás:** Egy űrlapot biztosít új járművek adatainak (gyártó, típus, motor, kategória, műszaki érv., gyártási év, leírás, ár) és képeinek feltöltésére. A POST kérést saját maga dolgozza fel. Több kép feltöltését is támogatja (`kep_url[]`). A képeket a szerveren a `/R-J_autoberles/kepek/` mappába menti egyedi névvel (`gyarto_tipus_szamlalo.ext`). A képek webes elérési útjait JSON tömbként tárolja a jarmuvek tábla `kep_url` mezőjében. Összekapcsolja a műszaki évet a szervizek tábla ID-jával. Visszajelzést ad session üzenetben.
 - **Listázás:** Lekérdezi és táblázatos formában megjeleníti az összes járművet az adatbázisból.

- **Törlés:** Minden sor mellett van egy "Törlés" gomb, amely egy modális ablakot nyit meg megerősítéshez. A tényleges törlést egy rejtett űrlap küldi el POST kéréssel. A törlés előtt tranzakciót indít, és törli a kapcsolódó rekordokat a berleksek, akciok, és velemenyek táblákból is a referenciális integritás megőrzése érdekében (vagy a hibák elkerülése végett). Ha minden törlés sikeres, commitálja a tranzakciót, egyébként rollback-et hajt végre.
 - **Módosítás:** Minden sor mellett van egy "Módosítás" gomb (link), amely az `autok_kezeles_modositas.php`-re irányít a jármű ID-jával GET paraméterként.
- **autok_kezeles_modositas.php:**
 - Lekérdezi a GET id alapján a módosítandó jármű adatait.
 - Megjelenít egy űrlapot, előre kitöltve a jármű adataival.
 - Lehetővé teszi a meglévő képek törlését (checkboxok) és újak feltöltését. Kezeli az elsődleges kép kiválasztását (rádiógombok).
 - A POST kérés feldolgozásakor validálja az adatokat. Kezeli a képtörlést (fizikai fájl törlése is), az új képek feltöltését (a meglévők mellé, egyedi névvel), és az elsődleges kép beállítását (a JSON tömb sorrendjének módosításával).
 - Frissíti a jármű adatait és a `kep_url` JSON mezőt a `jarmuvek` táblában (UPDATE). Visszajelzést ad session üzenetben, beleértve az esetleges képkezelési hibákat is. Visszairányít saját magára (PRG minta).
- **Adatbázis:** `jarmuvek`, `felhasznalas`, `szervizek`, `berleksek`, `akciok`, `velemenyek`.
- **3.2.4. Jogosultságkezelés** (`admin_jogosultsag.php`, `admin_jog_mod.php`)
 - **Leírás:** Felhasználók listázása, jogosultságuk (admin/normál) módosítása, adataik módosítása és törlése.
 - **Megvalósítás:**
 - **admin_jogosultsag.php:**
 - **Jogosultság Módosítás:** Egy űrlapot biztosít, ahol legördülő listából kiválasztható a felhasználó és beállítható az admin státusza (1 vagy 0). A POST kérést saját maga dolgozza fel, UPDATE utasítással módosítja a `felhasznalo` tábla `admin` mezőjét.
 - **Listázás:** Táblázatban megjeleníti az összes regisztrált felhasználót adataikkal (név, email, cím, pontok, admin státusz).

- **Törlés:** Minden sor mellett van egy "Törlés" gomb (modális megerősítéssel), amely POST kérést küld (rejtett űrlapon keresztül) a felhasználó törlésére a felhasznalo táblából. **Figyelem:** Ez a törlés potenciálisan sértheti a referenciális integritást, ha a felhasználóhoz tartoznak bérlekések vagy vélemények, és nincs ON DELETE CASCADE beállítva az adatbázisban! (A járműtörlésnél ez kezelve van, itt látszólag nincs.)
 - **Módosítás:** Minden sor mellett van egy "Módosítás" gomb (link), ami az admin_jog_mod.php-re irányít a felhasználó nevével (felhasznalo_nev) GET paraméterként.
- **admin_jog_mod.php:**
 - Lekérdezi a GET id (felhasználónév) alapján a felhasználó adatait.
 - Űrlapot jelenít meg a felhasználó adatainak (név, email, cím, hűségpontok) módosítására. A felhasználónév nem módosítható.
 - Lehetőséget ad új jelszó megadására (hasheléssel).
 - A POST kérést saját maga dolgozza fel, validálja az adatokat (email formátum, jelszó egyezés/hossz), majd UPDATE utasítással frissíti a felhasznalo táblát. Kezeli az esetleges email ütközést. Visszajelzés session üzenetben, átirányítás saját magára (PRG).
- **Adatbázis: felhasznalo.**
- **3.2.5. Bérlesek kezelése (admin_berlesek.php, admin_berlesek_mod.php)**
 - **Leírás:** Meglévő bérlekések listázása, újak manuális hozzáadása, meglévők módosítása és törlése.
 - **Megvalósítás:**
 - **admin_berlesek.php:**
 - **Hozzáadás:** Egy űrlapot biztosít új bérlet manuális rögzítésére (jármű kiválasztása, bérlő kiválasztása név alapján, dátumok megadása). A POST kérést saját maga dolgozza fel, validálja az adatokat (létező felhasználó/jármű, dátumok), majd INSERT utasítással beszúrja a berlesek táblába. Nem számol/ad hűségpontot, és a fizetési státusz alapértelmezetten nincs beállítva (vagy NULL).
 - **Listázás:** Lekérdezi és táblázatban megjeleníti az összes bérletet, összekapcsolva a jarmuvek és felhasznalo táblákkal a nevek megjelenítéséhez. Kiírja a fizetési státuszt is ('helyszínen' vagy 'utalással').

- **Törlés:** Minden sorhoz "Törlés" gomb (modális megerősítéssel), amely POST kéréssel törli a rekordot a berlesek táblából.
 - **Módosítás:** Minden sorhoz "Módosítás" gomb (link), amely az admin_berlesek_mod.php-re irányít a berles_id-val GET paraméterként.
 - **admin_berlesek_mod.php:**
 - Lekérdezi a GET id alapján a bérlet adatait.
 - Űrlapot jelenít meg, ahol módosítható a jármű, a bérlet (felhasználónév), a dátumok és a fizetési státusz (legördülőből).
 - A POST kérést saját maga dolgozza fel, validálja az adatokat, majd UPDATE utasítással frissíti a berlesek táblát. Visszajelzés session üzenetben, átirányítás saját magára (PRG).
 - **Adatbázis:** berlesek, jarmuvek, felhasznalo.
- **3.2.6. Véleménykezelés (admin_velemenyek.php)**
 - **Leírás:** A felhasználók által beküldött vélemények (járműspecifikus és általános) listázása, törlése és megválaszolása.
 - **Megvalósítás:**
 - **Listázás:** Lekérdezi és táblázatban megjeleníti az összes véleményt a velemenyek táblából, összekapcsolva a jarmuvek táblával (ha van jarmu_id) a jármű nevének kiírásához. Megjeleníti az esetleges admin választ is.
 - **Törlés:** Minden sorhoz "Törlés" gomb (modális megerősítéssel), amely POST kéréssel törli a rekordot a velemenyek táblából.
 - **Válaszadás:** Minden sorhoz tartozik egy szövegmező és egy "Válasz" gomb. A gombra kattintva POST kérést küld saját magának. A szkript UPDATE utasítással beírja az admin választ az admin_valasz mezőbe a velemenyek táblában. Ezt követően lekérdezi a véleményt író felhasználó email címét a felhasznalo táblából, és PHPMailer segítségével elküldi neki az admin választ emailben. Visszajelzést ad session üzenetben.
 - **Adatbázis:** velemenyek, jarmuvek, felhasznalo.
 - **Külső szolgáltatás:** SMTP szerver (PHPMailer).
- **3.2.7. Akciókezelés (admin_akciok.php)**
 - **Leírás:** Akciók (kedvezmények adott járművekre adott időszakban) létrehozása, listázása és törlése. Külön funkció Black Friday akció gyors létrehozására.

- **Megvalósítás:**
 - **Hozzáadás:** Egy űrlap, ahol kiválasztható a jármű, megadható a kedvezmény százaléka, a kezdő és záró dátum, egy rövid leírás, és bepipálható, ha Black Friday akcióról van szó. A POST kérést saját maga dolgozza fel, INSERT utasítással beszúrja az akciók táblába.
 - **Black Friday Hozzáadás:** Egy egyszerűbb űrlap, ahol csak a járművet kell kiválasztani. A POST kérés feldolgozásakor fixen 50% kedvezményt, az aktuális év november 28-30 közötti dátumokat, "Black Friday..." leírást és is_black_friday = 1 értéket állít be, majd INSERT-tel beszúrja.
 - **Email Értesítés:** Mindkét hozzáadási művelet után lekérdezi az összes felhasználó email címét, és PHPMailerrel tömeges emailt küld az új akció részleteiről.
 - **Listázás:** Lekérdezi és táblázatban megjeleníti az összes akciót az akciók táblából, összekapcsolva a jarmuvek táblával a jármű nevéért.
 - **Törlés:** Minden sorhoz "Törlés" gomb (modális megerősítéssel), amely POST kéréssel törli az akciót az akciók táblából.
- **Adatbázis:** akciók, jarmuvek, felhasznalo.
- **Külső szolgáltatás:** SMTP szerver (PHPMailer).

3.3. API Végpontok (Profil és Bérletek)

- **3.3.1. Felhasználói Adatok (api/user.php)**
 - **Leírás:** Kezeli a felhasználói profiladatok lekérdezését (GET) és módosítását (POST).
 - **Megvalósítás:**
 - **GET:** Ellenőrzi a sessiont (\$_SESSION['felhasznalo_nev']). Ha érvényes, lekérdezi a bejelentkezett felhasználó adatait (név, email, cím, jogsi dátum, pontok, admin státusz) a felhasznalo táblából. JSON formátumban adja vissza az adatokat. Ha nincs session vagy nem található a user, 401/404 hibakódot és hibaüzenetet ad vissza JSON-ban.
 - **POST:** Ellenőrzi a sessiont. Fogadja a POST adatokat (név, email, cím, jogsi dátum, új jelszó, jelszó megerősítés). Validálja az adatokat (jelszóegyeztetés). UPDATE utasítással frissíti a felhasználó adatait a felhasznalo táblában (kivéve a jelszót). Ha új jelszó is érkezett és valid, azt hasheli (password_hash) és külön UPDATE utasítással frissíti a jelszo mezőt. JSON választ ad a sikerességről vagy hibáról.
 - **Adatbázis:** felhasznalo (SELECT, UPDATE).
 - **Válasz formátum:** JSON.

- **3.3.2. Bérleti Adatok (api/rentals.php)**
 - **Leírás:** Lekérdezi a bejelentkezett felhasználó korábbi bérleteinek adatait.
 - **Megvalósítás:** Ellenőrzi a sessiont. Lekérdezi a berlesek táblából azokat a rekordokat, amelyek a bejelentkezett felhasználóhoz (felhasznalo_nev) tartoznak. Összekapcsolja a jarmuvek táblával a járműadatokért (gyártó, típus, motor, napi ár). Kiszámolja a teljes bérleti összeget ((DATEDIFF(b.ig, b.tol) + 1) * j.ar). Az eredményeket JSON tömbként adja vissza. Ha nincs session, 401 hibakódot és üzenetet ad.
 - **Adatbázis:** berlesek, jarmuvek (SELECT).
 - **Válasz formátum:** JSON.

4. Adatbázis Tervezés

4.1. Adatbázis Séma (autoberles.sql)

Az alkalmazás egy relációs adatbázist használ (MySQL/MariaDB), amelynek sémáját az autoberles.sql fájl definiálja. Ez tartalmazza a táblák létrehozását (CREATE TABLE) és néhány mintaadat beszúrását (INSERT INTO).

4.2. Táblák Leírása

- **4.2.1. felhasznalo**
 - **Cél:** Felhasználói fiókok adatainak tárolása.
 - **Kulcsoszlopok:**
 - **felhasznalo_nev (PRIMARY KEY, VARCHAR):** Egyedi felhasználónév.
 - **nev (VARCHAR):** Teljes név.
 - **emailcim (VARCHAR):** Email cím (lehet UNIQUE index rajta).
 - **jogositvany_kiallitasDatum (DATETIME):** Jogosítvány lejáratát (vagy kiállítás?).
 - **szamlazasi_cim (VARCHAR):** Számlázási cím.
 - **jelszo (VARCHAR):** Hashelt jelszó.
 - **admin (TINYINT):** Jogosultsági szint (1=admin, 0 vagy NULL=normál).
 - **Telefonszám (VARCHAR):** Telefonszám (úgy tűnik, nincs használva?).
 - **husegpontok (INT):** Gyűjtött hűségpontok.
- **4.2.2. jarmuvek**
 - **Cél:** A bérelhető járművek adatainak tárolása.
 - **Kulcsoszlopok:**
 - **jarmu_id (PRIMARY KEY, INT, AUTO_INCREMENT):** Egyedi járműazonosító.

- **felhasznalas_id** (FOREIGN KEY -> felhasznalas, INT): Jármű kategóriája.
- **szerviz_id** (FOREIGN KEY -> szervizek, INT): Kapcsolat a szerviz/műszaki adatokhoz.
- **gyarto** (VARCHAR): Gyártó neve.
- **tipus** (VARCHAR): Modell neve.
- **motor** (VARCHAR): Motor leírása.
- **gyartasi_ev** (DATE): Gyártás dátuma.
- **leiras** (VARCHAR/TEXT): Részletes leírás.
- **ar** (INT): Napi bérleti díj.
- **kep_url** (MEDIUMTEXT/TEXT): A jármű képeinek webes elérési útjait tartalmazó JSON tömb.

- **4.2.3. felhasznalas**

- **Cél:** Járműkategóriák (felhasználási módok) tárolása.
- **Kulcsoszlopok:**
 - **felhasznalas_id** (PRIMARY KEY, INT): Kategória azonosítója.
 - **nev** (VARCHAR): Kategória neve (pl. Városi, Családi).

- **4.2.4. szervizek**

- **Cél:** Műszaki vizsga lejárati évek tárolása (?). Egyszerű összekötő tábla.
- **Kulcsoszlopok:**
 - **id** (PRIMARY KEY, INT, AUTO_INCREMENT): Egyedi azonosító.
 - **muszaki_vizs_lejarat** (INT): Az évszám, amikor a műszaki lejár.

- **4.2.5. berleksek**

- **Cél:** A bérleti tranzakciók rögzítése.
- **Kulcsoszlopok:**
 - **berles_id** (PRIMARY KEY, INT, AUTO_INCREMENT): Egyedi bérletazonosító.
 - **jarmu_id** (FOREIGN KEY -> jarmuvek, INT): A bérelt jármű azonosítója.
 - **felhasznalo** (FOREIGN KEY -> felhasznalo, VARCHAR): A bérlő felhasználóneve.
 - **tol** (DATE): Bérlet kezdeti dátuma.
 - **ig** (DATE): Bérlet végi dátuma.
 - **kifizetve** (TINYINT): Fizetési státusz (0=helyszínen, 1=utalva).

- **4.2.6. velemenyek**

- **Cél:** Felhasználói vélemények tárolása.
- **Kulcsoszlopok:**
 - **velemeney_id (PRIMARY KEY, INT, AUTO_INCREMENT):** Egyedi véleményazonosító.
 - **felhasznalo_nev (VARCHAR):** A véleményt író felhasználóneve. (Lehetne FK a felhasznalo táblára).
 - **uzenet (TEXT):** A vélemény szövege.
 - **datum (DATETIME):** A vélemény írásának időpontja.
 - **admin_valasz (TEXT):** Az admin válasza (opcionális).
 - **jarmu_id (INT):** Annak a járműnek az ID-ja, amelyről a vélemény szól (0 vagy NULL, ha általános). (Lehetne FK a jarmuvek táblára).

- **4.2.7. akciók**

- **Cél:** Járművekre vonatkozó akciók és kedvezmények tárolása.
- **Kulcsoszlopok:**
 - **akcio_id (PRIMARY KEY, INT, AUTO_INCREMENT):** Egyedi akcióazonosító.
 - **jarmu_id (FOREIGN KEY -> jarmuvek, INT):** Az akciós jármű azonosítója.
 - **kedvezmeny_szazalek (DECIMAL):** A kedvezmény mértéke százalékban.
 - **kezdete (DATE):** Akció kezdete.
 - **vege (DATE):** Akció vége.
 - **leiras (VARCHAR):** Akció leírása (opcionális).
 - **is_black_friday (TINYINT):** Jelzi, ha speciális Black Friday akció (1=igen, 0=nem).

4.3. Kapcsolatok

- **jarmuvek** kapcsolódik felhasználas-hoz (felhasznalas_id).
- **jarmuvek** kapcsolódik szervizek-hez (szerviz_id).
- **berlesek** kapcsolódik jarmuvek-hez (jarmu_id).
- **berlesek** kapcsolódik felhasználó-hoz (felhasznalo -> felhasznalo_nev).
- **akciók** kapcsolódik jarmuvek-hez (jarmu_id).
- **velemenyek** kapcsolódhat jarmuvek-hez (jarmu_id).
- **velemenyek** kapcsolódhat felhasználó-hoz (felhasznalo_nev).

5. Fájlstruktúra (Feltételezett)

/R-J_autoberles/

|-- admin/

| |-- php/

| | |-- admin_akciok.php

| | |-- admin_berleksek.php

| | |-- admin_berleksek_mod.php

| | |-- admin_jogosultsag.php

| | |-- admin_jog_mod.php

| | |-- admin_velemenyek.php

| | |-- autok_kezeles.php

| | |-- autok_kezeles_modositas.php

| | |-- adatLekeres.php # (Lehet, hogy ez globális és a fő php/ mappában van?)

| | `-- db_connection.php # (Lehet, hogy ez globális és a fő php/ mappában van?)

| `-- admin_favicon.png

|-- api/

| |-- rentals.php

| `-- user.php

|-- css/

| |-- admin.css

| |-- forum.css

| |-- husegpontok.css

| |-- index.css

| |-- jarmuvek.css

| |-- kapcsolat.css

| `-- style.css

|-- js/

| |-- index.js

| `-- jarmuvek.js

|-- kepek/ # Képek tárolási helye

| `-- (feltöltött képek...)


```
|-- php/
| |-- auto_adatok.php
| |-- cseveges.php
| |-- forum.php
| |-- husegpontok.php
| |-- index.php
| |-- jarmuvek.php
| |-- kapcsolat.php
| |-- login.php      # (Vagy az index.php kezeli)
| |-- logout.php
| |-- modal.php
| |-- modosit_profil.php
| |-- profilom.php
| |-- register.php
| |-- adatLekeres.php  # Globális adatlekérő függvények
| |-- db_connection.php  # Globális adatbázis kapcsolat
| |-- src/            # PHPMailer könyvtár
| | |-- Exception.php
| | |-- PHPMailer.php
| | |-- `-- SMTP.php
| |-- `-- vendor/      # Composer függőségek (ha használták)
| |-- `-- autoload.php
|-- node_modules/      # Bootstrap forrás (opcionális, lehet CDN is)
| |-- bootstrap/
| |-- `-- ...

|-- autoberles.sql      # Adatbázis séma és dump
|-- favicon.png

`-- test_full_application.py # Teszt szkript

content_copydownload

Use code with caution.
```

6. Felhasználói Felület (UI/UX)

6.1. Stíluslapok (style.css, admin.css, stb.)

- A style.css definiálja az általános megjelenést (színek, betűtípusok, elrendezés, animációk). Egy modern, letisztult dizájnrá törekszik, szürke és sötét árnyalatok dominálnak (#e8ecef, #1c2526). Animációkat használ a felhasználói élmény javítására (pl. fadeInBody, slideInUp).
- A Bootstrap 5 keretrendszert használja a reszponzív grid rendszerhez, alapvető komponensekhez (kártyák, modális ablakok, űrlapok, gombok) és utility class-okhoz.
- Külön CSS fájlok (forum.css, jarmuvek.css, stb.) tartalmazhatnak oldal-specifikus stílusokat.
- Az admin.css az adminisztrációs felület egyedi stílusait tartalmazza, valószínűleg egyszerűbb, funkcionálisabb megjelenést biztosítva.

6.2. Reszponzivitás

A Bootstrap grid rendszerének és a style.css-ben definiált media query-knek köszönhetően az oldal reszponzív, alkalmazkodik a különböző képernyőméretekhez (desktop, tablet, mobil). Mobilon a navigációs menü egy "hamburger" ikon mögé rejthető (menu-toggle). A kártyák elrendezése a képernyő szélességétől függően változik (pl. 4 oszlopból 1 oszlopba).

6.3. Kliensoldali Interakciók (index.js, jarmuvek.js)

- JavaScriptet használ a felhasználói élmény javítására:
 - Modális ablakok: index.js tartalmazza az openModal és closeModal függvényeket a bejelentkezési, regisztrációs és üzenetablakok dinamikus megjelenítéséhez/elrejtéséhez.
 - Szűrő panel: jarmuvek.js tartalmazhat logikát a szűrő panel lenyitásához/bezárásához (toggleFilterBtn).
 - Bérleti Modal: jarmuvek.js (vagy inline script a jarmuvek.php-ban) kezeli a bérleti modal megnyitását és az adatok (pl. dátumok) átadását.
 - Karakter számláló: Az auto_adatok.php és kapcsolat.php textarea mezőinél a hátralévő karakterek számát jeleníti meg.
 - Galéria: Az auto_adatok.php-n JavaScript kezeli a képgaléria modális ablakának megnyitását, bezárását, a képek közötti lapozást (gombokkal és billentyűzettel).
 - Profil oldal: profilom.php fetch API hívásokkal tölti be dinamikusan a felhasználói és bérleti adatokat.
 - Profil módosítás: modosit_profil.php fetch-csel tölti be és küldi el az űrlap adatait.
 - Admin Törlés Megerősítés: Az admin oldalakon JavaScript kezeli a törlés megerősítő modális ablakát és továbbítja a törlési kérelmet a megfelelő ID-val.

7. Biztonság

- **7.1. Autentikáció és Autorizáció:**
 - Session-alapú autentikáció (`$_SESSION['felhasznalo_nev']`).
 - Admin felület védelme session ellenőrzéssel (`$_SESSION['admin'] == true`).
 - Az API végpontok (`api/user.php`, `api/rentals.php`) is ellenőrzik a sessiont.
- **7.2. Adatbázis Biztonság (SQL Injection):**
 - Az alkalmazás következetesen használ prepared statementeket (`MySQLi prepare()`, `bind_param()`, `execute()`) az SQL injection támadások kivédésére az adatbázis-műveleteknél (`INSERT`, `UPDATE`, `DELETE`, `SELECT WHERE`). Ez a legfontosabb védelmi vonal.
- **7.3. Adatkezelés (Input Sanitization, XSS):**
 - Felhasználói inputok feldolgozásakor `trim()` és `filter_input()` / `htmlspecialchars()` függvényeket használ a felesleges szóközök eltávolítására, alapvető tisztításra és a Cross-Site Scripting (XSS) támadások kockázatának csökkentésére a kimenetben. Az `api.php`-ban és a módosító oldalakon `FILTER_SANITIZE_*` konstansok is megjelennek.
- **7.4. Jelszókezelés:**
 - Jelszavak tárolása hashelve történik a `password_hash()` függvénnyel (`PASSWORD_DEFAULT` algoritmus, ami követi az ajánlott gyakorlatot).
 - Bejelentkezéskor a `password_verify()` ellenőrzi a jelszót.
- **7.5. Fájlfeltöltés Biztonsága:**
 - Az admin járműkezelő (`autok_kezeles.php`, `autok_kezeles_modositas.php`) ellenőrzi a feltöltött fájlok MIME típusát (`mime_content_type` vagy `finfo`) és méretét.
 - Egyedi, tisztított fájlneveket generál a felülírás és a potenciálisan rosszindulatú fájlnevek elkerülése érdekében.
 - Ellenőrzi a célmappa létezését és írhatóságát.
- **7.6. További Megfontolások:**
 - **CSRF:** A módosító oldalakon (`autok_kezeles_modositas.php`, `admin_jog_mod.php`) van egy alapvető ellenőrzés, ami összeveti a GET és POST ID-kat/neveket. Ez nyújt némi védelmet, de egyedi, munkamenetenként generált CSRF tokenek használata robusztusabb lenne.
 - **Session Security:** Nincs explicit session biztonsági beállítás (pl. `session.cookie_httponly`, `session.use_strict_mode`) a kódban, ezeket a `php.ini`-ben vagy a kód elején kellene konfigurálni.

- **Hibakezelés:** A kód néhol `die()`-t használ kritikus hibáknál, ami nem ideális éles környezetben. Konzisztensebb hibakezelés (pl. naplózás és felhasználóbarát hibaoldal) javasolt. Az `error_log` használata jó gyakorlat.
- **Jogosultságok:** Az admin rendszer csak egy szintű (admin/nem admin). Komplexebb rendszerekben finomabb jogosultságkezelésre lehet szükség.
- **Directory Traversal:** A képkezelésnél a fájlnevek tisztítása és az elérési utak ellenőrzése fontos a directory traversal sebezhetőségek elkerülése érdekében.

8. Telepítés és Konfiguráció

1. **Webszerver:** PHP támogatással rendelkező webszerver (Apache, Nginx).
2. **PHP:** Megfelelő PHP verzió (valószínűleg 7.4+), engedélyezett MySQLi kiterjesztéssel és fileinfo kiterjesztéssel (kép MIME típus ellenőrzéshez).
3. **Adatbázis:** MySQL vagy MariaDB szerver.
4. **Adatbázis létrehozása:** Hozzon létre egy adatbázist (pl. `autoberles`) és importálja bele az `autoberles.sql` fájlt.
5. **Kapcsolat beállítása:** Módosítsa a `db_connection.php` fájlt (és esetleg az `admin/php/db_connection.php`-t, ha külön van) a helyes adatbázis szerver névvel, felhasználónévvel, jelszóval és adatbázisnévvel.
6. **Fájlok másolása:** Másolja a projekt fájljait a webszerver megfelelő dokumentumgyökér könyvtárába (pl. `htdocs/R-J_autoberles`).
7. **PHPMailer:** Ha a `vendor/autoload.php` nincs használatban, győződjön meg róla, hogy a `php/src/` mappa tartalma elérhető és a `require` útvonalak helyesek a `jarmuvek.php` és `admin_akciok.php` fájlokban. Konfigurálja az SMTP adatokat (felhasználónév, jelszó, host, port) ezekben a fájlokban.
8. **Jogosultságok:** Győződjön meg róla, hogy a webszervernek van írási joga a `kepek/` könyvtárhoz.

9. Tesztelés (`test_full_application.py`)

A projekt tartalmaz egy Python alapú end-to-end teszt szkriptet (`test_full_application.py`), amely a `requests` könyvtárat használja.

- **Célja:** Automatikusan teszteli a webalkalmazás főbb funkcióit HTTP kérések küldésével és a válaszok ellenőrzésével (státuszkód, tartalmi részletek).
- **Lefedettség:** Teszteli a főoldal elérhetőségét, regisztrációt, bejelentkezést, járművek oldalát, bérlést, véleményküldést, kapcsolatfelvételi űrlapot, hűségpont oldalt, fórumot, profil oldalt, kijelentkezést, valamint az összes adminisztrációs felület elérhetőségét (jármű-, bérlés-, vélemény-, felhasználó-, akciókezelés).
- **Futtatás:** Python környezet szükséges a `requests` könyvtár telepítésével (`pip install requests`). A szkriptet parancssorból kell futtatni (`python test_full_application.py`).
- **Megjegyzések:**

- A teszt feltételezi, hogy az alkalmazás a `http://localhost/R-J_autoberles` címen fut. Ezt szükség esetén módosítani kell (`base_url`).
- A tesztnek szüksége van érvényes session ID-kra a bejelentkezett felhasználói (`session_cookie`) és admin (`admin_cookie`) tesztekhez. Ezeket manuálisan kell beszerezni (pl. böngésző fejlesztői eszközökkel) és behelyettesíteni a szkriptbe.
- A tesztekben szereplő ID-kat (pl. `jarmu_id: 1`) és várt szövegeket (pl. "Sikeres regisztráció") lehet, hogy az aktuális adatbázis állapothoz és a PHP szkriptek pontos kimenetéhez kell igazítani.
- Ez egy alapvető end-to-end teszt, nem helyettesíti a részletesebb unit vagy integrációs teszteket.

10. Jövőbeli Fejlesztési Lehetőségek

- **Frontend Framework:** Modern JavaScript keretrendszer (React, Vue, Angular) bevezetése a dinamikusabb és interaktívabb felhasználói felület érdekében.
- **API:** Teljeskörű REST API kialakítása a frontend és backend szétválasztásához.
- **Online Fizetés:** Integráció valamilyen fizetési szolgáltatóval (pl. Stripe, Barion) az online fizetés lehetővé tételéhez.
- **Részletesebb Keresés/Szűrés:** Több szempontú (pl. felszereltség, váltó típusa), összetettebb szűrési és rendezési lehetőségek a járművek listájában.
- **Jelszó Visszaállítás:** "Elfelejtett jelszó" funkció implementálása biztonságos tokenekkel és email validációval.
- **Szerepkörök:** Részletesebb jogosultságkezelés (pl. szerkesztő, menedzser) az admin felületen.
- **Képkezelés Fejlesztése:** Automatikus képátméretezés, vízjelezés, CDN használata.
- **Naplózás:** Részletesebb és strukturáltabb naplózás implementálása (események, hibák).
- **Tesztelés:** Unit és integrációs tesztek írása PHPUnit segítségével a kódminőség és megbízhatóság növelése érdekében.
- **Biztonság:** CSRF tokenek implementálása, session kezelés szigorítása, rate limiting bevezetése.
- **Aszinkron Feladatok:** Hosszabb műveletek (pl. tömeges email küldés) háttérfolyamatba szervezése (pl. üzenetsorral).
- **Felhasználói Élmény:** Dátumválasztó komponens javítása, AJAX alapú űrlapküldések.

11. Összegzés

Az R&J Autókölcsonzó egy funkcionálisan gazdag, PHP alapú webalkalmazás, amely sikeresen megvalósítja a kitűzött célokat mind a felhasználói, mind az adminisztrátori oldalon. Az alkalmazás alapvető biztonsági intézkedéseket (prepared statements,

password hashing) alkalmaz. A kód struktúrája logikus, külön fájlokkal az egyes főbb funkciókhoz és egy elkülönített admin résszel. A Bootstrap keretrendszer használata reszponzív és egységes megjelenést biztosít. Az api/ végpontok és a kliensoldali JavaScript javítják a felhasználói élményt a profil oldalon és más interaktív elemeknél. A projekt tartalmaz egy alap end-to-end teszt szkriptet is. A dokumentációban felsorolt jövőbeli fejlesztési lehetőségek mentén a rendszer tovább bővíthető és modernizálható. A projekt sikeresen demonstrálja a két fejlesztő (Jurcsó János Árpád - Felhasználói, Dávid Roland - Admin) együttműködését egy közös cél elérésében.