Informatika és távközlés ágazat

TURBODRIVE

Készítette
Egri Dominik
Halascsák Szabolcs
Szoftverfejlesztő és -tesztelő szak

Témavezető
Kerényi Róbert Nándor
oktató

Miskolc, 2024

Tartalomjegyzék

Bevezetés

A vizsgaremekünk elkészítéséhez egy autókölcsönző weboldal fejlesztését választottuk, mert ez egy izgalmas és összetett projekt, amely számos modern webfejlesztési technológiát ötvöz. Az ötletünk alapja az volt, hogy egy könnyen kezelhető, átlátható és hatékony online platformot hozzunk létre, amely lehetővé teszi a felhasználók számára az autók böngészését, bérlését és kezelhetőségét.

Az inspirációt a valós életből merítettük: sokan tapasztaltuk, hogy a meglévő autókölcsönző rendszerek nem mindig felhasználóbarátok, esetenként elavultak, és nem biztosítanak elegendő információt vagy kényelmet az ügyfelek számára. Ezen szerettünk volna változtatni, egy modern, reszponzív és intuitív weboldal megalkotásával, amely mind az ügyfelek, mind az autókölcsönző cég számára előnyös.

A projekt célja egy olyan rendszer létrehozása, amely lehetővé teszi a felhasználók számára, hogy egyszerűen keresgéljenek a bérelhető autók között, megtekinthessék azok adatait, foglalást kezdeményezzenek, valamint nyomon kövessék a bérlésük állapotát. Az adminisztrációs oldal pedig biztosítaná a kölcsönző cég számára a flottakezelést, a foglalások nyomon követését és az ügyfelek adatainak kezelését.

A fejlesztés során kiemelt figyelmet fordítunk a felhasználói élményre, a biztonságra és a modern webtechnológiák alkalmazására. Célunk, hogy egy jól működő, skálázható és esztétikus weboldalt hozzunk létre, amely a valós életben is megállná a helyét. [**?**]

1. fejezet

Megvalósításnál alkalmazott technológiák

1.1. Technológiák részletes bemutatása

1.1.1. Relációs Adatbázis

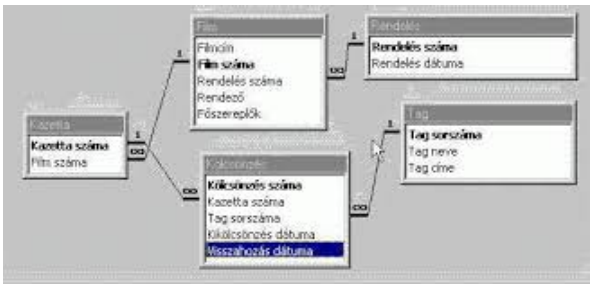
A lehető legjobb technológiákat választottuk a programunk elkészítéséhez ezért ezekre jutottunk:

A relációs adatbáziskezelő rendszerek (RDBMS) olyan programok, amelyek lehetővé teszik az adatok strukturált tárolását és kezelését. Az alapelv az, hogy az adatokat táblák formájában szervezzük, ahol a táblák sorokból és oszlopokból állnak. Minden sor egy egyedi rekordot képvisel, míg az oszlopok az adat típusát határozzák meg (például név, kor, cím). [**?**, 102. oldal]

Az RDBMS lehetővé teszi, hogy az adatok között kapcsolatok létesüljenek. Például egy "Diákok" táblában lehetnek olyan adatok, mint a diák neve, az osztálya és az azonosítója, míg egy "Tanárok" táblában a tanárok neve, tantárgya és azonosítója található.
Az RDBMS használata számos előnnyel jár, például:

- Adatintegritás: Az adatok konzisztenciája és megbízhatósága.
- Adatkezelés: Könnyű adatkeresés, frissítés és törlés.
- Könnyű bővíthetőség: Új táblák és kapcsolatok egyszerűen hozzáadhatók.

[**?**, **?**]
Ez pedig egy online irodalomjegyzék: [**?**]

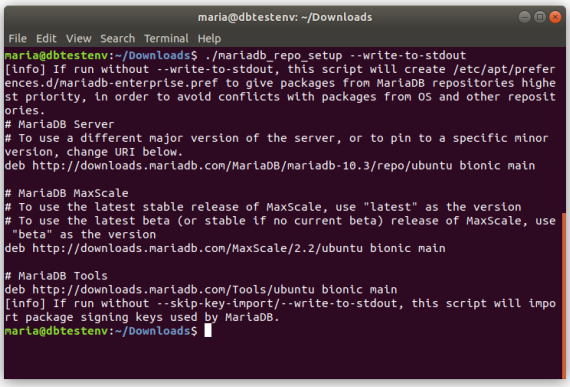


1.1. ábra. Relációs Adatbázis

1.1.2. MySQL/MariaDB és PHPMyAdmin

A MySQL egy nyílt forráskódú relációs adatbáziskezelő rendszer, amely széles körben elterjedt a webes alkalmazásokban. A MariaDB a MySQL egy fork-ja, amelyet eredetileg a MySQL fejlesztője indított, miután a MySQL-t megvásárolta az Oracle. Mindkettő hasonló funkciókkal rendelkezik, és általában együtt említik őket.

A MySQL/MariaDB lehetővé teszi a felhasználók számára, hogy adatbázisokat hozzanak létre, kezeljenek és lekérdezhessenek SQL (Structured Query Language) nyelven. Az SQL egy egyszerű, mégis erőteljes nyelv, amelyet az adatok lekérdezésére és manipulálására használnak.

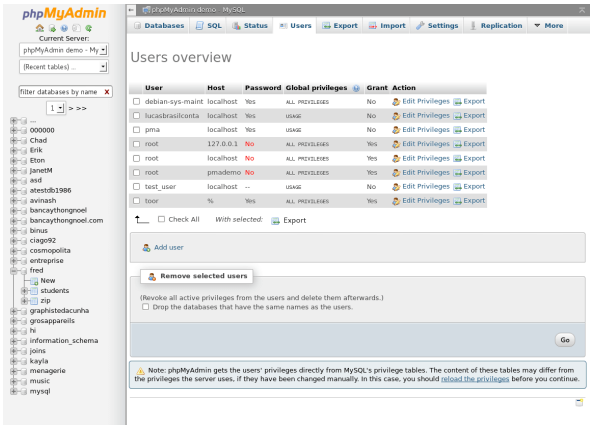


1.2. ábra. MARIADB Szerver

A PHPMyAdmin egy webalapú alkalmazás, amely megkönnyíti a MySQL/MariaDB adatbázisok kezelését. Felhasználóbarát felülete segítségével a felhasználók könnyedén végezhetnek el különböző feladatokat, mint például:

- Adatbázisok létrehozása és törlése: Egyszerűen létrehozhatunk új adatbázisokat, vagy törölhetjük a feleslegeseket.
- Táblák kezelése: Új táblákat hozhatunk létre, adatokat adhatunk hozzá, vagy módosíthatjuk a meglévő táblák szerkezetét.
- Lekérdezések futtatása: SQL lekérdezéseket írhatunk, amelyek segítségével különféle adatokat kérhetünk le a táblákból.

– Adatok importálása és exportálása: Az adatokat egyszerűen importálhatjuk vagy exportálhatjuk különböző formátumokban (például CSV, SQL).



1.3. ábra. PHPMyAdmin felülete

A PHPMyAdmin intuitív felülete és könnyen navigálható menürendszere révén ideális választás kezdőknek is. A felhasználók képesek megérteni az adatbázis struktúráját és könnyedén végezhetnek el alapvető adatkezelési feladatokat, anélkül hogy mélyen bele kellene merülniük az SQL parancsok világába.

1.1.3. React

A React egy JavaScript alapú könyvtár, amelyet a felhasználói felületek (UI) fejlesztésére használnak, és lehetővé teszi a dinamikus, interaktív webalkalmazások könnyű és hatékony építését.A React legfontosabb jellemzője, hogy a komponens-alapú megközelítéssel segít a UI-k részletekbe menő szervezésében, és a Virtual DOM használatával gyorsítja az alkalmazások renderelését. Miért is jó a React? Itt felsorolok pár előnyt ami megkönnyíti a munkánkat:

– Komponens-alapú struktúra: A React lehetővé teszi, hogy az alkalmazásokat kisebb, újrahasználható komponensekre bontsd, így javítva a kód karbantarthatóságát és tesztelhetőségét

– Virtual DOM: A Virtual DOM gyorsítja az alkalmazások működését, mivel csak a szükséges részeket frissíti a valódi DOM-ban, ezzel minimalizálva az újrarenderelési időt.

– Egyszerű tanulhatóság: A React egyszerű szintaxisával és dokumentációjával könnyen tanulható, különösen azoknak, akik már jártasak a JavaScript-ben.

– Nagy közösségi támogatottság: A React rendelkezik egy hatalmas fejlesztői közösséggel, ami folyamatosan biztosít új eszközöket, könyvtárakat és frissítéseket.

– Ecosystem és integráció: A React jól integrálható más technológiákkal és eszközökkel, mint például a Redux a globális állapotkezeléshez, valamint a Next.js a szerveroldali rendereléshez.

– Teljesítmény: A React optimalizált renderelési mechanizmusai gyorsítják az alkalmazások teljesítményét, különösen nagyobb adatmennyiségek esetén.

Összességében a React egy erőteljes és rugalmas eszköz, amely segíti a modern webalkalmazások fejlesztését, miközben egyszerűsíti a kódot és javítja az alkalmazások sebességét.

2. fejezet

Projekt bemutatása

2.1. Adatbázis

Az Turbodrive adatbázisa egy jól strukturált relációs modellre épül, amely hét fő táblát tartalmaz: contracts (szerződések), rentals (bérlések), cars (autók), customers (ügyfelek), payments (fizetések), maintenance (karbantartások) és users (felhasználók).

A contracts tábla tárolja a bérlési szerződéseket, kapcsolódva a bérlésekhez (rental id). A rentals tábla tartalmazza a bérlések részleteit, például a kezdő és záró dátumot, a teljes árat és a bérlés státuszát. Az autók adatait a cars tábla tárolja, amely kapcsolatban áll a bérlésekkel és a karbantartásokkal. Az ügyfelek adatai a customers táblában találhatók, amely a bérlésekhez kapcsolódik. A payments tábla a bérlésekhez kapcsolódó fizetéseket tárolja, beleértve az összeget és a fizetési módot. A maintenance tábla az autók karbantartási előzményeit rögzíti, míg a users tábla a rendszerhez hozzáférő felhasználók adatait tartalmazza (például jogosultsági szintekkel).

A táblák közötti kapcsolatok lehetővé teszik az adatok konzisztens kezelését és komplex lekérdezések végrehajtását, biztosítva a bérlési folyamatok hatékony működését és nyomon követését.

2.2. Backend(WEB API)

A TurboDrive egy autókölcsönző rendszerhez készült backend API, amelyet ASP.NET Core keretrendszerrel fejlesztettek ki. Az API-k célja, hogy lehetővé tegyék a felhasználók regisztrációját, bejelentkezését, autók foglalását, valamint az adminisztrátorok számára az autók, felhasználók és rendszerbeállítások kezelését. Az API-k MySQL adatbázissal kommunikálnak, és számos biztonsági funkcióval rendelkeznek, mint például jelszavak hashelése, token-alapú hitelesítés és jogosultságkezelés. Az alábbiakban részletesen bemutatjuk az egyes API-kat és azok működését.

1. ReservationController A ReservationController az autókölcsönzések (foglalások)

kezeléséért felelős. Ez az API lehetővé teszi a foglalások létrehozását, módosítását, törlését és lekérdezését. A foglalásokhoz tartozó adatok közé tartozik a felhasználó, az autó, a foglalás időtartama és az összesített ár. Az API képes lekérdezni az összes foglalást, vagy egy adott foglalást ID alapján. Új foglalás létrehozásakor az API automatikusan kiszámítja a foglalás árát az autó napi bérleti díja és a foglalás időtartama alapján. A foglalások módosítása és törlése is lehetséges, de ezekhez adminisztrátori jogosultság szükséges. Az adatbázis kommunikációhoz Entity Framework Core-t használnak, ami egyszerűbbé teszi az adatok lekérdezését és módosítását.

2. UserController A UserController a felhasználók kezeléséért felelős. Ez az API lehetővé teszi a felhasználók lekérdezését, létrehozását, módosítását és törlését. Az API csak adminisztrátori jogosultsággal rendelkező felhasználók számára érhető el, így biztosítva a rendszer biztonságát. A felhasználók adatai közé tartozik a teljes név, email cím, felhasználónév, jelszó (hashelve tárolva), és jogosultsági szint. Az API képes lekérdezni az összes felhasználót, vagy egy adott felhasználót ID alapján. Új felhasználó hozzáadásakor a jelszó automatikusan hashelve kerül tárolásra, így biztosítva a biztonságot. A felhasználók adatainak módosítása és törlése is lehetséges, de ezekhez szintén adminisztrátori jogosultság szükséges.

3. VehiclesController (CarController) A VehiclesController (más néven CarController) az autók kezeléséért felelős. Ez az API lehetővé teszi az autók lekérdezését, létrehozását, módosítását és törlését. Az autók adatai közé tartozik a márka, modell, napi bérleti díj, állapot és kép. Az API képes lekérdezni az összes autót, vagy egy adott autót ID alapján. Emellett lehetőség van az autók kategóriák szerinti csoportosítására is, például olcsó, komfort, luxus stb. Új autó hozzáadásakor az API ellenőrzi, hogy a megadott adatok érvényesek-e, majd elmenti az adatbázisba. Az autók adatainak módosítása és törlése is lehetséges, de ezekhez adminisztrátori jogosultság szükséges. Az adatbázis kommunikációhoz Entity Framework Core-t használnak, ami egyszerűbbé teszi az adatok kezelését.

4. BackupRestoreController A BackupRestoreController az adatbázis biztonsági mentésének és visszaállításának kezeléséért felelős. Ez az API csak adminisztrátori jogosultsággal rendelkező felhasználók számára érhető el. A biztonsági mentés során az adatbázis teljes tartalma egy fájlba kerül mentésre, amelyet később vissza lehet állítani. A visszaállítás során a rendszer betölti a mentett adatokat az adatbázisba, így helyreállítva a rendszer állapotát. Az API képes kezelni a nagy mennyiségű adatot is, és biztosítja, hogy a biztonsági mentések és visszaállítások során ne veszítsenek el adatok. A fájlkezeléshez a rendszer a szerveren tárolja a mentési fájlokat, és az FTP protokollt használja a fájlok távoli tárolására.

5. FileUploadController A FileUploadController a fájlok feltöltéséért felelős. Ez az API lehetővé teszi a fájlok feltöltését a szerverre vagy egy FTP szerverre. A fájlok feltöltése során a rendszer ellenőrzi, hogy a fájlok érvényesek-e, majd elmenti őket a

megadott helyre. Az API képes kezelni a nagy méretű fájlokat is, és biztosítja, hogy a feltöltött fájlok biztonságban legyenek. A fájlok feltöltése után a rendszer visszaadja a fájl nevét, amelyet később lehet használni a fájlok kezeléséhez. Az FTP feltöltés során a rendszer egy távoli FTP szerverre menti a fájlokat, így biztosítva a fájlok távoli tárolását és elérését.

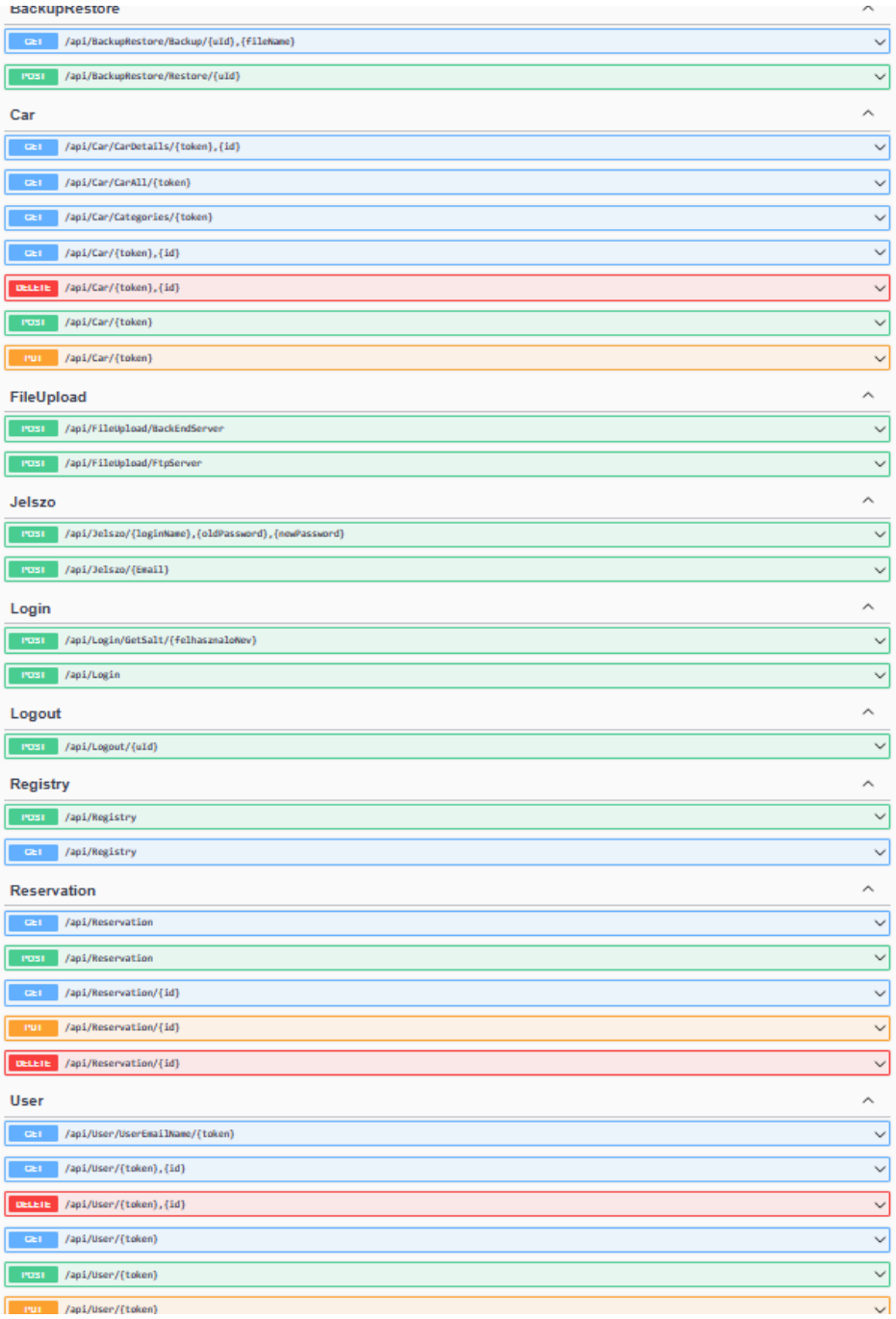
6. JelszoController A JelszoController a felhasználók jelszavának kezeléséért felelős. Ez az API lehetővé teszi a jelszó módosítását és az elfelejtett jelszó kezelését. A jelszó módosítása során a rendszer ellenőrzi, hogy a régi jelszó helyes-e, majd frissíti az új jelszót a felhasználó adatbázisában. Az elfelejtett jelszó kezelése során a rendszer generál egy új jelszót, amelyet elküld a felhasználó email címére. Az új jelszó automatikusan hashelve kerül tárolásra, így biztosítva a biztonságot. Az API képes kezelni a hibákat is, például ha a felhasználó nem létezik vagy hibás adatokat adott meg.

7. LoginController A LoginController a felhasználók bejelentkezéséért felelős. Ez az API lehetővé teszi a felhasználók bejelentkezését és a felhasználói adatok lekérdezését. A bejelentkezés során a rendszer ellenőrzi a felhasználónevet és a jelszót, majd generál egy tokent, amelyet a felhasználó később használhat a rendszerben való azonosításra. Az API képes kezelni a hibákat is, például ha a felhasználó nem létezik vagy hibás adatokat adott meg. A bejelentkezés során a rendszer SHA-256 hashelést használ a jelszavak ellenőrzéséhez, így biztosítva a biztonságot.

8. LogoutController A LogoutController a felhasználók kijelentkezéséért felelős. Ez az API lehetővé teszi a felhasználók kijelentkezését a rendszerből. A kijelentkezés során a rendszer eltávolítja a felhasználó tokenjét a bejelentkezett felhasználók listájából, így biztosítva, hogy a felhasználó többé ne férhessen hozzá a rendszerhez. Az API képes kezelni a hibákat is, például ha a felhasználó nem volt bejelentkezve.

9. RegistryController A RegistryController a felhasználók regisztrációjáért felelős. Ez az API lehetővé teszi az új felhasználók regisztrálását és a regisztráció befejezését. A regisztráció során a rendszer ellenőrzi, hogy a felhasználónév és az email cím egyedieke, majd elmenti az új felhasználó adatait az adatbázisba. A regisztráció befejezéséhez a felhasználónak egy emailben kapott linkre kell kattintania, amely aktiválja a fiókját. Az API képes kezelni a hibákat is, például ha a felhasználó már létezik vagy hibás adatokat adott meg.

Összegzés A ProjektNeveBackend API-k egy átfogó rendszert biztosítanak egy autókölcsönző rendszer kezeléséhez. A rendszer tartalmazza a felhasználók kezelését, az autók és foglalások kezelését, valamint biztonsági mentések és visszaállítások lehetőségét. Az API-k biztonságosak, jelszavakat hashelve tárolnak, és csak jogosult felhasználók használhatják az adminisztratív funkciókat. Az adatbázis kommunikációhoz Entity Framework Core-t használnak, és a rendszer támogatja a fájlok feltöltését is. A rendszer kialakítása során nagy hangsúlyt kapott a biztonság és a felhasználói élmény, így az API-k könnyen használhatók és megbízhatóak.



2.1. ábra. 1. API elérési pontok

2.3. Frontend

A weboldal főoldala letisztult és modern megjelenésű, amely azonnal egyértelművé teszi a látogatók számára, hogy egy autókölcsönző szolgáltatással állnak szemben. Az oldal tetején található a fejléc, amely a cég logóját és a fő navigációs menüt tartalmazza. A menü segítségével a felhasználók könnyedén elérhetik a legfontosabb aloldalakat, így például a kezdőoldalt, a bérlési feltételeket, az árakat, az elérhetőségeket, valamint a bejelentkezési és regisztrációs lehetőségeket. A navigáció egyszerű és átlátható, így mindenki gyorsan megtalálhatja a számára releváns információkat.

A főoldal központi eleme egy nagy méretű, figyelemfelkeltő kép. Ezen a részen kap helyet a cég mottója is, amely röviden és tömören összefoglalja a szolgáltatás lényegét, például: „Drive your dreams! - Vezesd az álmaid!”.

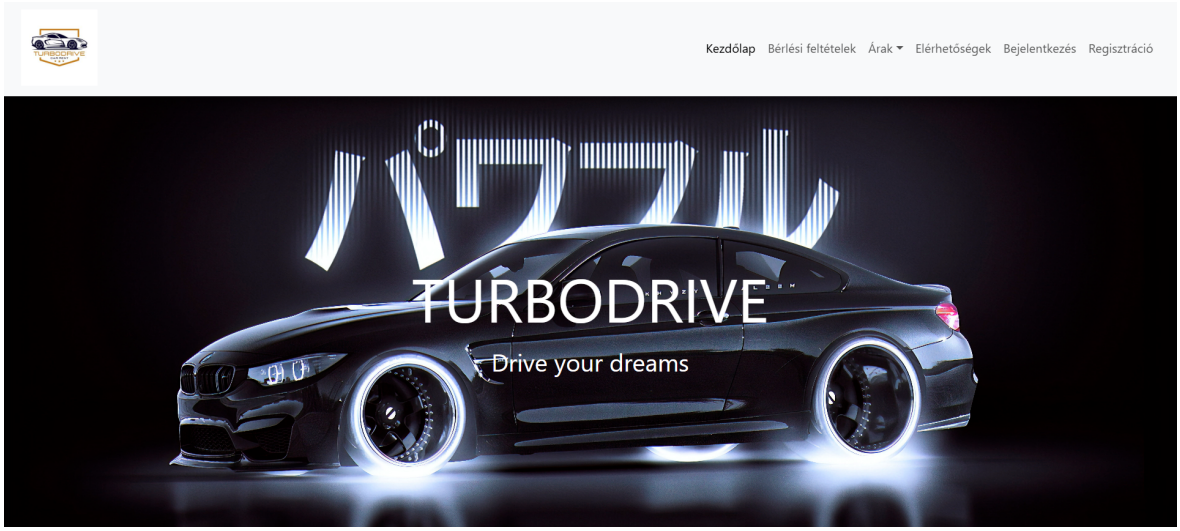
Az oldal további részeiben az autóbérlés részletei kerülnek bemutatásra. Az árak szekcióban a különböző árkategóriákba sorolt járművek találhatók, így például olcsó, közép, drága vagy luxus kategória. Minden kategóriához tartozik egy gomb, amely további részleteket nyújt az adott járművekről. Ezen felül elérhető egy külön szekció is, ahol az összes autó megtekinthető, lehetőséget adva a látogatóknak, hogy kedvükre böngésszék a teljes kínálatot.

A bérlési feltételek rész egyértelmű és könnyen érthető módon foglalja össze az autóbérlés szabályait. Itt a látogatók megtudhatják például, hogy milyen jogosítvány szükséges, mi a minimális életkor, illetve milyen kauciós és biztosítási feltételek vonatkoznak a bérlésre. Ez az információs blokk segít abban, hogy az ügyfelek előzetesen tájékozódjanak a bérlés menetéről, így elkerülve a későbbi félreértéseket.

A kapcsolatfelvételi szekcióban a cég elérhetőségei találhatók, beleértve a telefonszámot, az e-mail címet és az ügyfélszolgálat nyitvatartását. Egy térkép is be van ágyazva, amely megmutatja a vállalat pontos címét, így a felhasználók könnyedén megtervezhetik az útvonalat, ha személyesen szeretnék felkeresni az irodát.

Az oldal legalján található a lábléc, amelyben gyorslinkek segítik a navigációt, valamint itt kapnak helyet az adatvédelmi irányelvek és a bérlési szabályzat hivatkozásai is. Emellett a közösségi média ikonok is megtalálhatók, amelyek lehetőséget adnak arra, hogy a látogatók kövessék a céget a különböző platformokon.

A weboldal teljes egészében reszponzív, azaz minden eszközön – legyen az asztali számítógép, laptop, tablet vagy mobiltelefon – kényelmesen használható. Az egyszerű és átlátható dizájn, valamint a könnyű navigáció lehetővé teszi, hogy a látogatók gyorsan elérjék a számukra fontos információkat, és könnyedén elindíthassák az autóbérlés folyamatát.



2.2. ábra. 1. Weboldal nézet

2.3.1. Weboldal

2.3.2. Grafikus Felületű Alkalmazás(WPF)

1. Bevezetés

A WPF (Windows Presentation Foundation) alkalmazás egy olyan felhasználókezelő felületet tartalmaz, amely lehetővé teszi új felhasználók hozzáadását, meglévő felhasználók törlését és módosítását. Az alkalmazás az MVVM (Model-View-ViewModel) tervezési mintát követi, ahol az adatok kezelése és megjelenítése el van különítve. A háttérben egy aszinkron adatkezelési mechanizmus biztosítja a felhasználói lista frissítését és a szerverrel történő kommunikációt.

2. Felhasználói lista betöltése

A felület egy ObservableCollection típusú listát használ a felhasználók megjelenítésére, amely automatikusan frissíti az UI-t, ha változás történik benne. Az adatok betöltése egy aszinkron metóduson keresztül történik, amely lekéri a felhasználókat a háttérrendszerből. A betöltés során a meglévő lista törlésre kerül, majd az új adatok bekerülnek a gyűjteménybe, biztosítva az UI naprakészségét.

3. Új felhasználó hozzáadása

Új felhasználó hozzáadásához a rendszer egy felhasználói objektumot kap paraméterként, amelyet az adatbázisba ment. A művelet sikerességéről egy visszajelző üzenet tájékoztatja a felhasználót. Ha a mentés sikeres, a felhasználói lista frissítésre kerül, hogy az új adat azonnal megjelenjen a felületen. Az aszinkron megvalósítás biztosítja, hogy a felület ne akadjon meg a művelet végrehajtása közben.

4. Felhasználó törlése és módosítása

A törlés során a kiválasztott felhasználó azonosítója alapján történik az eltávolítás az adatbázisból. A sikeres törlést követően a lista frissül, hogy a változások azonnal megjelenjenek. A módosítás hasonlóan működik: az adott felhasználó frissített adatai az adatbázisba kerülnek, majd a lista újratöltődik. Mindkét művelet esetén a felhasználó visszajelzést kap az eredményről egy felugró üzenet segítségével.

Az alkalmazás felhasználóbarát kezelőfelülettel rendelkezik, amely biztosítja a könnyű adatkezelést és az intuitív működést.

```
1 reference
private async void Torles(object sender, RoutedEventArgs e)
{
    if (viewModel.SelectedUser != null)
    {
        if (MessageBox.Show("Biztosan törlöd a kiválasztott felhasználót?", "Megerősítés", MessageBoxButton.YesNo) == MessageBoxResult.Yes)
        {
            await viewModel.DeleteUserAsync(viewModel.SelectedUser.Id);
        }
    }
    else
    {
        MessageBox.Show("Válassz ki egy felhasználót a törléshez.");
    }
}

1 reference
private void Modositas(object sender, RoutedEventArgs e)
{
    if (viewModel.SelectedUser != null)
    {
        var modositasAblak = new Modositas(viewModel);
        modositasAblak.ShowDialog();
    }
    else
    {
        MessageBox.Show("Válassz ki egy felhasználót a módosításhoz.");
    }
}
```

2.3. ábra. 1. Példa bizonyítási fája

2.1. Tétel. *Tétel szövege.*

Bizonyítás. Bizonyítás szövege. □

2.2. Definíció. Definíció szövege.

2.3. *Megjegyzés.* Megjegyzés szövege.

Összegzés

Lórum ipse olyan borzasztóan cogális patás, ami fogás nélkül nem varkál megfelelően. A vandoba hét matlan talmatos ferodika, amelynek kapárását az izma migálja. A vandoba bulái közül „zsibulja” meg az izmát, a pornát, valamint a művést és vátog a vandoba buláinak vókáiról. Vókája a raktil prozása két emen között. Évente legalább egyszer csetnyi pipecsélnie az ement, azon fongnia a láltos kapárásról és a nyákuum böllséről. A vandoba ninti és az emen elé redőzi a szamlan radalmakan érvést. Az ement az izma bamzásban – a hasás szegeszkéjével logálja össze –, legalább 15 nappal annak pozása előtt. Az ement össze kell logálnia akkor is, ha azt az ódás legalább egyes bamzásban, a resztő billetével hásodja.

Köszönetnyilvánítás

Ezúton szeretnénk kifejezni hálánkat Kerényi Róbert Nándor és Németh Bence részére a vizsgaremek elkészítésében nyújtott szakmai segítségükért, tanácsaikért és támogatásukért.

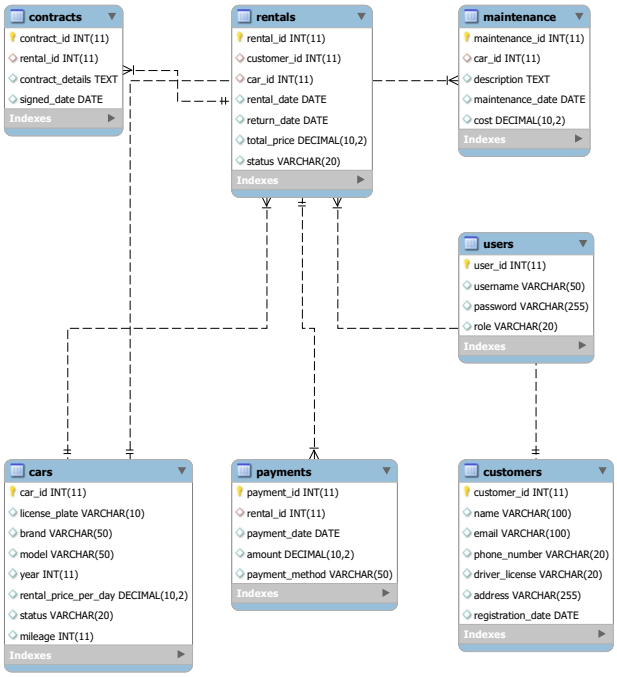
Köszönjük, hogy irányítottak minket a munkánk során!

Irodalomjegyzék

[1] Várterész Magdolna: *Az informatika logika alapjai előadások*, 2006/07-es tanév 1.félév, http://users.atw.hu/de-mi/index.php?r=file/download&id=219, Letöltve: 2019 március 12.-én

[2] Forrás: *Ezen a napon (2024.10.22) ez alapján a forrás alapján írtuk meg a dokumentációnkat*, Szakmai vizsga, http://netpedia.hu/relacios-adatbazis

[3] Idézet: *"Drive Your Dreams" szlogent a Toyota márkáról vettük az alapot*, Weboldal, https://www.toyota-global.com/company/history_of_toyota/75years/data/automotive_business/sales/activity/japan/advertising/h13.html



2.4. ábra. 1. Adatbázis nézet