

# VOLVO Cars Korporésön vizsgaremek

Készítette: Demeter Márton & Hegyes Áron

Osztály: 2/14sz.

Helyszín: Budapest

## Tartalom

VOLVO Cars Korporésön vizsgaremek.....	1
Bevezetés:.....	4
Miért pont ezt választottuk? .....	4
Milyen programokat használunk? .....	4
Mik lesznek a funkciók?.....	4
Felhasználói dokumentáció: .....	5
Hogyan telepítsük és indítsuk a programot? .....	5
A weboldal működése: .....	5
Admin felület: .....	8
Fejlesztői dokumentáció:.....	10
car-brands.php: .....	10
car-types.php:.....	11
cars.php: .....	11
clients.php: .....	12
connect.php:.....	12
dashboard.php: .....	12
index.php:.....	13
Minták: .....	14
footer.php:.....	14
header.php: .....	14
Felhasználói felület: .....	15
Adatbázis: .....	15
'cars' tábla: .....	15
'car_brands' tábla:.....	15
'car_types' tábla: .....	16
'clients' tábla: .....	16
'reservations' tábla:.....	16
'users' tábla: .....	17
'regusers' tábla: .....	17
Főoldal: .....	18
Kezdőlap szekció:.....	18
Szolgáltatások szekció:.....	18
Rólunk szekció: .....	18
Brendek szekció: .....	18
Kölcsönzés szekció: .....	18

Kölcsönzés szekció részletezése:.....	19
Session kezelése: .....	19
Űrlap feldolgozása: .....	19
Elérhető autók lekérdezése: .....	19
Foglalás végrehajtása: .....	19
Vue.js alkalmazás a validációhoz: .....	19
Modal ablakok a járműleírások megjelenítéséhez: .....	19
Bejelentkezés és Regisztráció: .....	20
userlogin.php:.....	20
Session kezelés, ezen kívül HTML fejléc: .....	20
Bejelentkezési adatok ellenőrzése:.....	20
Bejelentkezési űrlap:.....	20
userregister.php:.....	21
Regisztrációs adatok feldolgozása: .....	21
Regisztrációs űrlap:.....	21

## Bevezetés:

**Miért pont ezt választottuk?** Elsősorban azért, mert az egyikünk apja, épp pont a VOLVO-nál dolgozik. Innen jött az inspiráció. Mivel külföldön egyre jobban népszerű a VOLVO és azoknak a bérlete, ezért gondoltuk azt, hogy ez egy jó projekt lenne, egy ilyen weboldal létrehozása.

**Milyen programokat használunk?** Elsősorban Visual Studio Code lesz a főbb fejlesztői környezet. Az adatbázis részhez phpMyadmin, XAMPP lesz használatban. Programozási nyelvek közül értelemszerűen HTML, CSS és JavaScript lesz alap a frontend részhez. Backendhez pedig php-t használunk.

**Mik lesznek a funkciók?** Elsősorban az alap funkciók egyike lesz, hogy lehessen regisztrálni majd persze bejelentkezni. Ezek után átkerülünk a főoldalra, aminek egy rendkívül egyszerű, de annál szebb dizájnya van.

Első főbb funkció, amit lát az ember mikor belép az oldalra az egy navigációs sáv lesz, amivel az oldal bizonyos részeire lehet navigálni. Elsőre kevésbé feltűnő funkció a reszponzivitás. Következő fő funkció, amit lát majd a felhasználó az a kölcsönzési folyamat első része, ahol pár adat megadásával lehet a további részére menni a kölcsönzésnek. A dokumentáció írása alatt a weboldal 9 kölcsönözhető autóval rendelkezik. Ezek közül tud választani a felhasználó. Majd beírja a nevét, email-címét és telefonszámát, és a kölcsönzés végre is hajtható.

További rendkívül fontos funkció, hogy egy időpontban, vagy csak ha ütközik is egy időpont, egy ugyanolyan autót nem lehet majd bérelni.

Egy rendkívül fontos nem felhasználói funkció, pedig egy admin felület. Ezen belül pedig számos további funkció van. Ilyen például a kölcsönzések kezelése, új brandek hozzáadása, új kocsi típus hozzáadása, illetve új autó

**Mik a célok?** Elsősorban az, hogy sikeres projektet hozzunk létre :D. Másodszor pedig az, hogy egy felhasználóbarát, könnyen használható felületet hozzunk létre, amit bárki tud használni. Amennyiben sikeres lesz a projekt, lehet, hogy tovább is fogjuk fejleszteni. Több funkcióval például, nem csak egy autómárkára specifikált oldal. Szerintünk nagyon jó lenne, ha egy külön profin megcsinált mobilos applikáció is. És természetesen az lenne a legjobb, ha ténylegesen használnák az emberek, de ez még távoli terv.

## Felhasználói dokumentáció:

### Hogyan telepítsük és indítsuk a programot?


Elsősorban a XAMPP nevű program telepítésére van szükség. Majd pedig a programunk fájljait tartalmazó „volvokorporésön” nevű mappát elhelyezni a C:/xampp/htdocs helyre. Amint ezzel megvagyunk, indítani kell a phpMyAdmint. Ezt pedig úgy tudjuk megtenni, hogy először a xampp-ot indítjuk, azon belül pedig 2 modult kell elindítani. Apache és MySQL. Következő lépés pedig az admin gomb megnyomása a MySQL modul sorában. Ezzel át is kerülünk a phpMyAdmin felületére. Itt létre kell hozni egy „car\_rental” nevű adatbázist, amibe pedig importálni kell a mellékelt fájlban lévő SQL fájlt. Amint ezzel megvagyunk, és sikeresen importáltuk az adatbázist a program indításra kész. A böngésző keresőjébe az alábbi kell beírni: „localhost/volvokorporésön”. Ha minden sikeres volt már látható is a gyönyörű weboldal.

**A weboldal működése:** A funkcióknál már említett navigációs sáv lesz vélhetőleg az első amit a felhasználó észrevesz. Ezzel tud navigálni az oldalon.




A „Home” azaz a kezdőlap mindig az oldal legtetejére viszi vissza a felhasználót, ezzel könnyítve azt, hogy ne kelljen folyamatosan görgetni.


A „Services” azaz a szolgáltatások szekció pedig egy tipikus szövegekkel teli szekcióhoz vezeti a felhasználót. Itt van az igazi önfényezés, ezért jó a mi cégünk, kiváló szakemberek stb...

**Expert Technicians**


Experience the reliability of our car rental service, maintained by expert technicians for worry-free journeys.

**Professional Service**


Experience the epitome of professionalism with our car rental service, meticulously maintained by expert technicians for your utmost satisfaction.

**Great Support**


Benefit from outstanding support with our car rental service, backed by a team of skilled technicians ensuring a seamless experience.

**Technical Skills**

Rely on our car rental service, supported by unmatched technical skills to guarantee a high-performance and reliable driving experience.

**Highly Recommended**

Discover excellence with our highly recommended car rental service, praised for its exceptional quality and customer satisfaction.

**Positive Reviews**

Explore our car rental service, celebrated for its positive reviews, reflecting the satisfaction of our customers and the quality of our offerings.

A „Brands” azaz a brendek pedig egy kisebb részhez navigál minket, ahol lehet látni azokat a Volvo márkacsoportokat, amik közül lehetséges a bérlet. Ennek különösebb funkciója nincs, tényleg csak annyi, hogy megmutatja a választható autók csoportjait.



A következő szekció a „Reserve” azaz a kölcsönzés. Ez talán az egyik legfontosabb része az egész oldalnak. Itt indul meg a kölcsönzés folyamata. A bérlet felhasználatának ki kell tölteni némi adatot. Ilyen az, hogy hol szeretnénk felvenni, majd hol szeretnénk leadni az autót. A dokumentáció írása során ez még opcionális, bármilyen cím írható. További ki kell választani, hogy milyen napon szeretnénk felvenni, illetve leadni az autót.

### FIND YOUR CAR

Pickup Location

Váci út 21

Return Location

Váci út 21

Pickup Date

é.é.é.é. hh. nn.

Return Date

é.é.é.é. hh. nn.

Book Instantly

Majd, ha rányomtunk a „Book Instantly” gombra akkor a felhasználó átkerül egy másik felületre, ahol kiválasztja a 9 autó egyikét, hogy melyiket szeretné kibérelni. Majd pedig megadja a teljes nevét, illetve az email-címét, majd a telefonszámát. Eddig még nem került be semmi az adatbázisba, egészen addig amíg a „Book It Now” gombra kattintunk. Ekkor kapunk egy megerősítést, a foglalás sikerességéről. Az adatok pedig bekerültek

az adatbázisba, illetve ezáltal pedig az admin oldal frissítése után, már itt is látható a kölcsönzés. Ezután vissza is térhetünk a főoldalra. Ha ismét kölcsönözni szeretnénk, és ugyanazokat az adatokat adjuk meg (kölcsönzés ideje, felvétel/leadás helye), akkor azt megtehetjük, viszont a korábban választott autót már nem lehet bérelni, mivel az már bérelve van. Ilyenkor szimplán nem jelenik meg a választási lehetőségek között az adott autó. Ez természetesen fontos, hogy ne legyen ütközés a foglalásoknál, például 2 külön személy ugyanazon hétre foglalta ugyanazt az autót.

A foglalás utolsó fázisának a felülete pedig az alábbi módon néz ki:

Pickup Date : 2024-04-07

Return Date : 2024-04-08

Pickup Location : Váci Út 21

Return Location : Váci Út 21

Volvo S40	Select
Volvo S60	Select
Volvo S90	Select
Volvo V40	Select
Volvo V60	Select
Volvo V90	Select
Volvo XC40	Select
Volvo XC60	Select
Volvo XC90	Select

Full Name

Feles Elek

E-mail

xyz@gmail.com

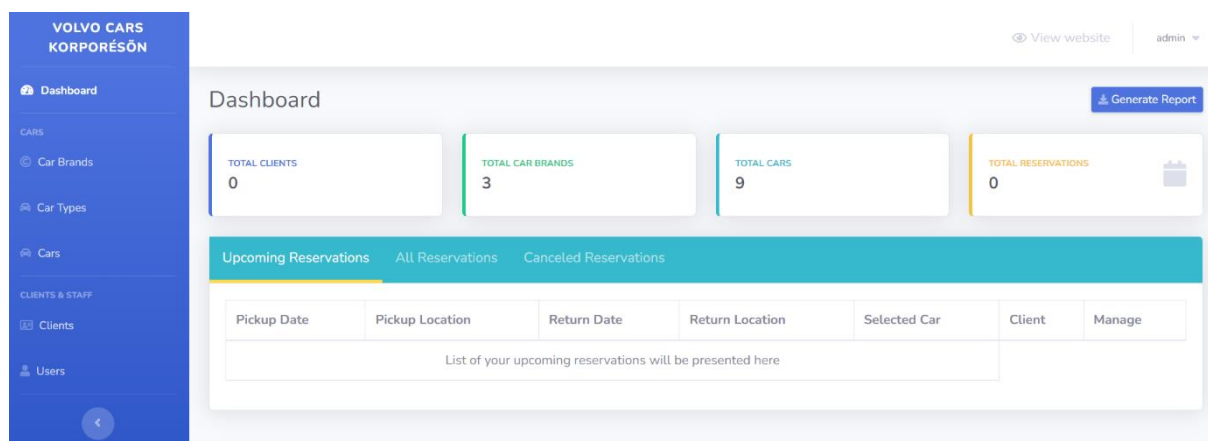
Phone numbder

0123456789

Book instantly.

Látható, hogy itt megjelennek a fontosabb adatok magáról a kölcsönzésről.

**Admin felület:** Egy ilyen oldal rendkívül fontos, kihagyhatatlan, ha egy igazán jó weboldalt szeretnénk létrehozni. Ez teljesen különálló rész, nem érhető el a fő weboldalról. Ebbe a felületbe kerülnek be az admin felhasználóknak a weboldalon megadott adatok. Ilyen például egy kölcsönzés, és annak adatai. Ebben a felületben lehet kezelni magukat a kölcsönzéseket, például törölni azokat. Itt van lehetőség új autótípusokat, brendeket hozzáadni. Ilyenkor az új adatok természetesen az adatbázisba kerülnek.



Látható a kliensek, brendeik, autók és a kölcsönzések száma is. Külön kategóriákra vannak bontva a kölcsönzések, ilyen az „Elkövetkezendő kölcsönzések”, „Összes kölcsönzések”, illetve a lezárt kölcsönzések.

Ez a felületünk rendelkezik számos funkcióval, ilyen például a brendeik kezelése. Ez egy

Brand ID	Brand Name	Brand Image	Manage
13	Volvo S Series		
14	Volvo V Series		
15	Volvo XC Series		


igaz könnyen használható része a felületünknek. Láthatóak az oldalon megjelenő brendeik, ami jelen esetben 3 darab található. Mindegyik brand rendelkezik egy ID-val, névvel és egy képpel. Új hozzáadás pedig rendkívül egyszerű. Kattintani kell a „+ Add New Brand” gombra, ami egy kis ablakot hoz fel, ahol megadjuk a brand nevét, például „Volvo EX Széria”, és kiválasztjuk a megfelelő képet. Lehetőségünk van

egy adott brand törlésére is. Ez nagyon egyszerűen úgy működik, hogy rámegyünk a „Manage” szekcióban lévő gombra, majd pedig meg kell erősíteni, hogy biztosan törölni szeretnénk e.



Az utóbbihoz nagyon hasonlóan működik a következő funkciónk. Itt tudunk autó típusokat hozzáadni az oldalunkhoz. Jelen esetben a szedán, kupé, és elektromos áll rendelkezésünkre. A típusok is rendelkeznek ID-val, típus névvel, illetve egy 1-1 mondatot, rövid, de lényegre törő leírással. Új típus hozzáadása nem működik másképp, mint a korábbiakban. Megadjuk a típus nevét, és a leírást, és hozzá is adódik.

Következő funkció, maguknak az új autóknek a hozzáadása. Frissen futtatott weboldalnál 9 autóval rendelkezünk. Mind a 3 brandhez 3-3-3 autó. Mindegyiknek van különálló ID, név, brand id, típus stb.... Úgy autó hozzáadásánál ezeket kell megadnunk. Egy kisebb formot kell kitölteni.

Cars							
+ Add New Car							
Car ID	Car Name	Brand ID	Car Type	Color	Model	Description	Manage
89	Volvo S40	13	1	White	S40	The Volvo S40 is a compact sedan that embodies Swedish engineering excellence. With seating for five passengers, it offers a comfortable and spacious interior. Known for its dynamic handling, the S40 provides an engaging driving experience, whether navigating city streets or cruising on the highway. Equipped with advanced safety features, it ensures peace of mind for both driver and passengers.	

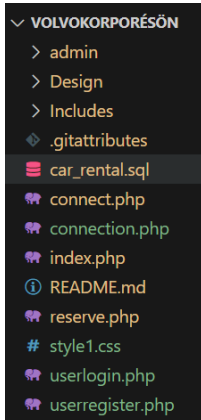
A leírások maximális hossza 1000 karakterben lett megadva, ezzel lehetővé téve azt, hogy egy kiváló információadagot adjunk át a felhasználónak. Jelen esetben kb. ennek a háromnegyedét érik el csak a leírások.

További két felület található meg az oldalon, de azok már nem igazán tartalmaznak funkciót. Egyik kis felületen az admin felhasználókat lehet látni, Jelen esetben ez nem is többszám, csak egy admin felhasználó van. Újat hozzáadni csak az adatbázisból közvetlenül tudunk.

Az utolsó kis felület pedig a kölcsönzések listája, ahol névvel, telefonszámmal, email-címmel láthatjuk a kölcsönzőt.

## Fejlesztői dokumentáció:

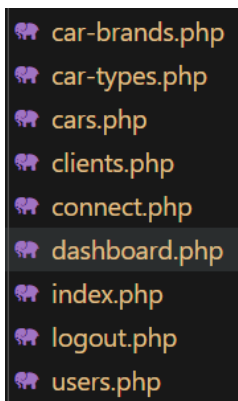
A teljes weboldal php, javaScript, css és persze SQL-re épül föl. Az oldal fő elemei/szekciói mind-mind egy külön php fájlból ered.



Az oldal mappaszerkezete első ránézésre igen egyszerűnek tűnhet. 3 mappa, és pár php fájl. Viszont ez sokkalta többet rejt magában.

Először is, az admin mappa: Az admin mappán belül további 3 mappa és 9 php fájl található. Először nézzük meg, hogy az adott php fájlok milyen feladatokat hajtanak végre.

### car-brands.php:



Ez a fájl, az autómárka kezelő felületet valósítja meg. Amikor a felhasználó, azaz az admin, belép az oldalra. Belépéskor session-ök segítségével ellenőrizzük, hogy az admin be van-e jelentkezve. Ha igen, akkor megjelenik a felület, egyébként pedig átirányítja a bejelentkezés felületére.

Az autómárkák listáját egy táblázatban jeleníti meg a felület, és ezáltal egy egyszerű, könnyen kezelhető oldalt kapunk. Minden autómárka sorában olvasható az azonosító, név és egy kép.

Egy új autómárka hozzáadásához az oldalon lévő megfelelő gombra kattintva egy űrlap jelenik meg. Itt kötelező megadnunk az autómárka nevét, és egy képet kell feltölteni. Amikor az admin elküldi a kitöltött űrlapot (másképp nem is engedi), a php kód feldolgozza az adatokat, és beilleszti őket az adatbázisba. Ha sikeres volt a feltöltés, kapunk egy megerősítést, ha pedig nem akkor hibaüzenet jelenik meg.

Minden autómárka mellett található egy törlés gomb, aminek a megnyomására egy megerősítő ablak jelenik meg, hogy biztosan szeretnénk-e törölni. Ha igen akkor az adott autómárka törlődik az adatbázisból, és a felületből is.

## car-types.php:

Ez a kód egyautómodell-specifikus interfészt valósít meg. Az előző kódhoz hasonlóan van egy bejelentkezési ellenőrzés, amely változatlan marad.

Az űrlap elküldésekor a kód ellenőrzi, hogy a POST kérés olyan-e, amely új típust ad hozzá. Ha igen, akkor feldolgozza az űrlap adatait, és beilleszti az adatbázisba. Ha a beillesztés sikeres vagy sikertelen, a megfelelő üzenet jelenik meg (a sikeres beillesztés megerősítése vagy hibaüzenet).

Ha a felhasználó az "Új típus hozzáadása" gombra kattint, egy modális ablak (egy már alaphoz betöltött felugró ablak, ami egy gomb megnyomására jelenik meg, és blokkolja az eredeti oldallal való interakciót) jelenik meg. Ebben az ablakban a felhasználó megadhatja és elküldheti az új típus adatait.

A típus törlése szintén a modális ablakban történik. Válassza ki a törölni kívánt típust a listából, és nyomja meg a megfelelő "Törlés" gombot. Megjelenik egy megerősítő ablak, ahol a felhasználó megerősítheti a törlést.

A Vue.js keretrendszer leegyszerűsíti az űrlapok érvényesítését. Figyeli az űrlap állapotát, és a beküldés előtt érvényesíti a beírt adatokat. Ha valami hiányzik vagy hibás, az űrlap nem kerül elküldésre.

## cars.php:

Ez a kód egy autókezelő rendszert mutat, ahol az autók hozzáadhatók és kezelhetők. Először a felhasználó bejelentkezését ellenőrzi a munkamenet ahogy eddig is. Ha a bejelentkezés sikeres, megjelenik az 'Autók' oldal.

Az oldal felépítése egyszerű, de annál átláthatóbb és hatékonyabb. A felhasználót egy szép fejléc fogadja, amely tartalmazza az oldal címét és egy "Jelentés létrehozása" gombot. Funkciója nincs a gombnak.

Az "Új autó hozzáadása" gombra kattintva egy modális ablak nyílik meg, ahol új autók adhatók hozzá az adatbázishoz. Az űrlap kitöltése után a beküldött adatokat feldolgozza és beilleszti az adatbázisba. A felhasználónak üzenet jelenik meg, ha a hozzáadás sikeres vagy sikertelen volt.

A járműadatok táblázatos formában jelennek meg. Minden járműazonosítójával, nevével, márkájával, színével, modelljével és leírásával jelenik meg. Ezenkívül a jármű minden sorban szerkeszthető vagy törölhető. Törlés esetén egy megerősítő modális ablak jelenik meg, amely lehetővé teszi a felhasználó számára, hogy megerősítse a törlési szándékát.

## clients.php:

A táblázatban szereplő adatok az ügyfelek teljes nevét, telefonszámát és e-mail címét tartalmazza, valamint egy egyedi azonosítót is. Az adatbázisból kerülnek lekérdezésre és megjelenítésre.

## connect.php:

Ez a kód csak azért felel, hogy a car\_rental nevű adatbázisunkhoz csatlakozzunk. Ez a kód rendkívül fontos, e nélkül nem is működne, hogy az adott részek ahogy kellene. Például nem jelennének meg az adatok.

## dashboard.php:

Ez talán a leghosszabb kód, ami nem css/javascript. Itt is számos olyan funkcióval találkozunk, amit már láttunk a korábbiakban, azt nem is részleteznénk.

Kölcsönzés törlése: Amikor az admin valamilyen okból törölni szeretne egy foglalást, akkor ez a kód a felelős. Elsősorban ellenőrizzük, hogy ki lett-e töltve a kis űrlap. Amennyiben igen, és a törlés sikeres volt, módosul az adatbázis, illetve a felületünkön is más helyre kerül a foglalás. Ehhez előkészítjük az SQL lekérdezést, amely az UPDATE parancsot használja a reservations táblában. Frissítjük a canceled mező értékét 1-re, amivel jelezzük a foglalás törlését, valamint beállítjuk a cancellation\_reason mezőt a megadott törlési indokra.

Amennyiben a művelet sikertelen, akkor természetesen egy hibaüzenet jelenik meg a felhasználónak.

Kontent sor: Ez a része a kódnak négy darab kártyát jelenít meg a felhasználó számára. Mindegyik egy külön statisztikát mutat meg.

Mindegyik kártya egy külön oszlopban helyezkedik el a felületen, és a bootstrap keretrendszer oszlopainak segítségével van elrendezve.

Minden kártya rendelkezik egy fejléccel, ami tartalmazza a statisztika nevét. Ezen kívül egy számláló látható a név alatt, ami megjeleníti a statisztika értékét (pl.: összes kölcsönzés). A szép és átlátható megjelenés érdekében mindegyik kártya külön színjelzéssel van ellátva.

Az adatok dinamikusan vannak betöltve a php kódban található 'countItem' függvénnyel. Ez számolja az adott táblában a rekordok számát, és a kellő értéket jeleníti meg a megfelelő kártyán.

Kölcsönzések tábla: Igazából ez nem is egy tábla, hanem egy kisebb felület, ami 3 táblát kezel. Válthatunk az „Upcoming” azaz a közelgő, az „All” azaz az összes és a „Canceled” azaz a törölt kölcsönzések táblája között.

A táblázatokban minden külön sor 1-1 kölcsönzést reprezentál és magának a foglalásnak az adatait tartalmazza. Ilyen például a kölcsönzés kezdetének és végének az ideje, a kiválasztott autó, a kölcsönző neve. Amennyiben nem történt még foglalás egy rövid üzenet jelenik csak meg a táblában, ami arról tájékoztat minket, hogy itt fognak megjelenni a kölcsönzések.

A kölcsönzés törlése gombra való kattintáskor egy a már korábbiakban említett modál ablak jelenik meg, ahol egy indokot kell megadnunk, és megerősíteni a törlési szándékot.

[index.php](#):

Ez a kód tartalmazza az admin felületbe való bejelentkezést. Az oldalra való belépés előtt itt is ellenőrzi, hogy van-e aktív session. Amennyiben van, átirányít minket a a dashboard.php oldalra.

Amennyiben nincs aktív session, akkor természetesen nem irányít át minket, hanem maradunk a bejelentkezés oldalán. Amennyiben helytelen a felhasználónév vagy a jelszó, egy hibaüzenet jelenik meg.

## Minták:

### footer.php:

Az oldal lábléce a footer elem, aminek háttérszíne fehér, és a tartalmat egy "container" osztályú elembe helyezi. A lábléc tartalmazza a „szerzői jogi” információt, ami a Volvo Cars Korporésön Rental Service 2023-ra vonatkozik.

Ezt követi a modál elem, ami a kijelentkezést biztosítja az admin(ok) számára. Ha az admin szeretne kijelentkezni, akkor megjelenik a felugró ablak, ami rákérdez, hogy biztosan ki szeretne-e lépni. Ez a logout.php fájlra irányít át, ami végrehajtja a kijelentkezést.

logout.php:

```
<?php
    session_start();

    session_unset();

    session_destroy();
    header('Location: index.php');
    exit();
?>
```

### header.php:

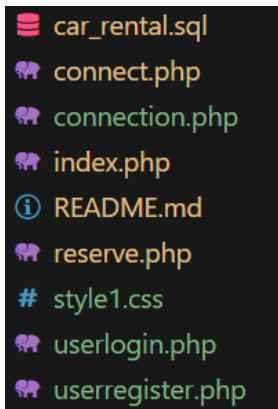
Az oldal fejlécét egy "wrapper" div elem veszi környékén, ami a Bootstrap keretrendszer "container-fluid" osztályát alkalmazza. Ez a div elem az oldal fő tartalomkonténere.

A fejléc tartalmazza az oldalsávot (sidebar), mely az oldal navigációs menüjét tartalmazza. A navigációs menü tartalmazza a Volvo Cars Korporésön logóját, továbbá különféle menüpontokat, mint példának okáért a Dashboard, Car Brands, Clients, Users stb.

A felhasználó nevének megjelenítéséhez, valamint a kijelentkezés lehetőségének biztosításához a fejléc jobb felső sarkában található egy felhasználói menü, mely lehetővé teszi a felhasználó részére a profiljának megtekintését, beállítások módosítását, tevékenységnapló megtekintését, ezen kívül a kijelentkezést.

## Felhasználói felület:

### Adatbázis:



A felhasználó felületet ez a fő elemek építik fel. Első ilyen, amit láthatunk, az a `car_rental.sql` fájl. Ez tartalmazza az összes tábla létrehozását, és azok feltöltését is a kellő adatokkal. Nem egy hatalmas adatbázis, összes 7 táblát tartalmaz, amik a következők: `users` (admin), `regusers` (regisztrált felhasználók), `clients` (akik már hajtottak végre kölcsönzést), `car_brands` (autó brendek), `car_types` (autó típusok), `cars` (autók) és végül a `reservations` (kölcsönzések).

### 'cars' tábla:

Ebben a táblában 7 mezőt definiáltunk. Az `'id'` egy egyedi azonosító, ami az autókat azonosítja a táblában. A `'car_name'` az csak szimplán az autó nevét tartalmazza. A `'brand_id'` az autó márkáját azonosító kulcs, ami a `'car_brands'` táblához kapcsolódik. A `'type_id'` a típust azonosító kulcs, ami a `'car_types'` táblához kapcsolódik. A `'color'` értelemszerűen az autó színét tartalmazza, a `'model'` pedig az évjáratot, hogy hámnyas modell, azt jelzi. A `'description'` pedig az autó hosszas leírását tartalmazza.

Az `'id'`, `'brand_id'` és a `'type_id'` `int` típusú adattípust használnak míg a `'car_name'`, `'color'`, `'model'` és a `'description'` szöveges, tehát `varchar` adattípust használ. Utóbbiaknak a maximális karakterhossza is definiálva van.

### 'car\_brands' tábla:

Ebben a táblában összesen csak 3 mező van definiálva. Ilyen a `'brand_id'`, `'brand_name'` és a `'brand_image'`. Az `'brand_id'` itt is, csak úgy, mint a `'cars'` táblában, egy egyedi azonosító az egyes brendeknek. A `'brand_name'` az a brend neve, és a `'brand_image'` pedig az adott brenchez tartozó képfájl elérési útja.

A `'brand_id'` mező szám értékeket tárol, így az `int` adattípust használ. A `'brand_name'` és a `'brand_image'` `varchar` adattípust használ, meghatározott hosszúságokkal.

### 'car\_types' tábla:

Ebben a táblában az előzőhöz hasonlóan szintúgy csak 3 mező van definiálva. Ez a 3 pedig a 'type\_id', 'type\_label' és a 'type\_description'. A 'type\_id' ahogy eddig is (és a továbbiakban is) egyedi azonosító. A 'type\_label' az autó típusának a neve (szedán, kupé, elektromos). A 'type\_description' pedig az autó részletesebb/hosszabb leírását tartalmazza.

A 'type\_id' int típusú adattípust használ, míg a 'type\_label' és a 'type\_description' varchar adattípust.

### 'clients' tábla:

Itt ahogy látható, már 4 mező van definiálva. 'client\_id', 'full\_name', 'client\_email' és végül 'client\_phone'.

A id kéréssel ez a tábla mindenhol varchar adattípust használ mivel itt csak szöveg típusú adatokat tárolunk. Ilyen a 'full\_name' ami a kölcsönző teljes nevét tárolja. A 'client\_email' a kölcsönző email elérhetőségét tárolja, míg végül a 'client\_phone' az ügyfél telefonszámát tárolja el.

### 'reservations' tábla:

A kölcsönzések tábla a legnagyobb az adatbázisunkban, mivel itt kell a legtöbb adatot eltárolni.

Az első ilyen a 'reservation\_id' ami egy egyedi azonosító. A 'client\_id' pedig magát a kölcsönző ügyfelet azonosító id, majdnem ugyanúgy, mint a 'car\_id'-nál, ahol pedig magának az autónak az azonosítója az id. A 'pickup\_date' és a 'return\_date' date, azaz dátum adattípust használ. Ez a két érték határozza meg, hogy a felhasználó mikor veheti fel, illetve mikor kell leadnia az autót. Hasonló a következő kettő mező, ami a 'pickup\_location' és a 'return\_location'. Itt annyi a különbség, hogy varchar adattípust használunk.

A 'canceled' mezőt tinyint(1) adattípus definiálja. Alapértelmezett értéke 0, tehát nem törölt kölcsönzés. Ha pedig 1, akkor az törölt. És végül a 'cancellation\_reason' az egy egyszerű varchar adattípust használ. Azt a szöveget tárolja el, amit az admin beírt, mint indoka a kölcsönzés törlésének.



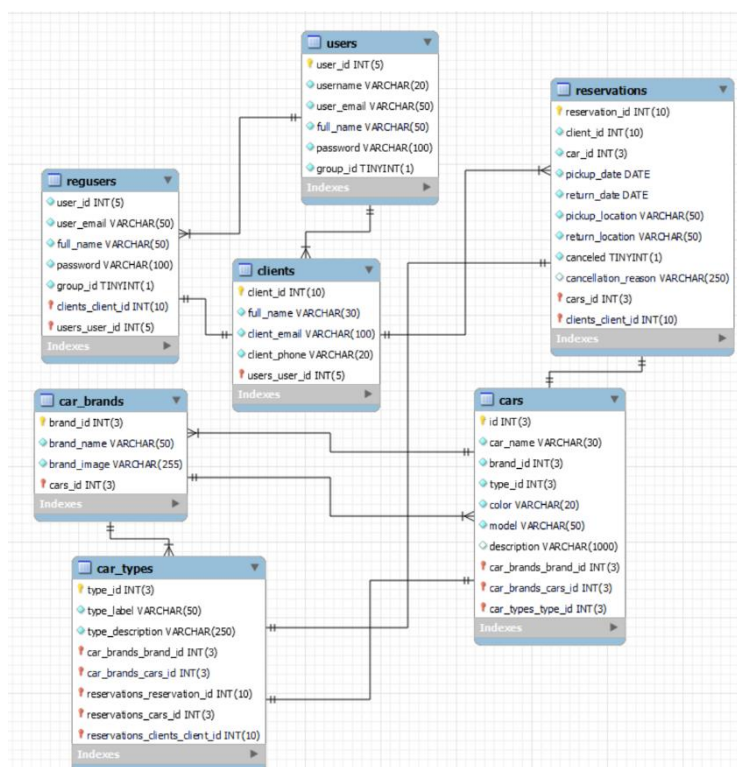
### 'users' tábla:

Ebben a táblában, ami valójában az admin felhasználókat tárolja, 6 mezőt definiálunk, amiből az első itt is az id.

A 'username' az admin felhasználónevét tárolja, tehát varchar adattípust használva. A 'user\_email' természetesen az admin email-címét tárolja, szintúgy varchar adattípust használva. Ez érvényes lesz a következő mezőkre is. A 'full\_name' a teljes nevét tárolja el, a 'password' pedig a jelszót tárolja el. És végül a 'group\_id' ami annyiban különbözik az eddigi id mezőktől, hogy ez egy csoportazonosító ami tinyint(1) adattípust használ.

### 'regusers' tábla:

Ez nem különbözik sokban a sima 'users' táblától. A különbség szimplán annyi, hogy nincs 'user\_name'. Minden más stimmel.



## Főoldal:

Ez az a felület, amit minden felhasználó lát. 5 jelentősebb részre bontható az egész oldal.

## Kezdőlap szekció:

A kezdőlap szekció az oldal első benyomásának kialakításáért felel. Itt található a weboldal címe, a szlogenek, valamint egy rövid bemutatkozó szöveg.

A HTML, továbbá CSS kód itt határozza meg a szekció vizuális megjelenését, például a betűtípust, a színeket, ezen kívül a elrendezést.

A PHP kód dinamikusan betölti a szövegeket, ezen kívül képeket az adatbázisból, lehetővé téve a könnyű frissítést, továbbá testreszabást a weboldal tulajdonosának.

## Szolgáltatások szekció:

Ez a szekció részletesen ismerteti a vállalkozás által kínált szolgáltatásokat. Itt találhatóak olyan elemek, mint az ügyfélszolgálat, a professzionális szolgáltatások, ezen kívül a technikai támogatás.

A HTML és CSS kód itt is felel a vizuális megjelenésért. Minden szolgáltatást egy-egy blokkban jelenít meg, amelyek tartalmát dinamikusan tölti be a PHP kód az adatbázisból.

## Rólunk szekció:

Ez a szekció részletesebben bemutatja a vállalkozás filozófiáját és történetét. Az oldalra való betekintést, ezen kívül kapcsolódást szolgálja.

A szekció vizuális megjelenését a HTML és CSS kódok határozzák meg. A szöveges tartalom, továbbá az illusztrációk dinamikusan betöltődnek a PHP kód segítségével az adatbázisból.

## Brendek szekció:

Ez a rész az általuk kínált autómárkákat mutatja be képekkel. Az autók márkanevvel, ezen kívül logóval vannak azonosítva.

A PHP kód segítségével az adatokat betölti az adatbázisból, majd egy foreach periódus segítségével jeleníti meg az összes márkát, valamint azok logóit.

## Kölcsönzés szekció:

Ez a szekció lehetőséget nyújt az autók közvetlen foglalására. A felhasználó megadhatja a kívánt foglalási, ezen kívül visszavételi időpontot, ezen kívül a helyszíneket.

A HTML űrlap a felhasználói felületet biztosítja a foglalási adatok megadására. A PHP kód a felhasználó által megadott adatokat ellenőrzi, valamint feldolgozza a megfelelő műveletek végrehajtásához.

## Kölcsönzés szekció részletezése:

A főoldal szekció közül ez az, ami még sokkalta tovább megy mint a többi, így ezt is számos részre lehet bontani.

## Session kezelése:

A `session_start()` függvény indítja el a munkamenetet, hogy tárolni lehessen az űrlapból származó adatokat.

Az `include` utasításokkal hozzáadjuk a szükséges fájlokat és funkciókat, például a kapcsolódást az adatbázishoz, továbbá a funkciókat az adatok ellenőrzéséhez.

## Űrlap feldolgozása:

Ha a felhasználó elküldi az űrlapot (`reserve_car` gomb lenyomásával), az adatokat a `$_POST` tömbből olvassuk ki, majd elmentjük őket a munkamenetben a későbbi felhasználás céljából.

## Elérhető autók lekérdezése:

Ha már kiválasztották a foglalási időpontokat (`$_SESSION` változók léteznek), akkor lekérdezzük az elérhető autókat az adatbázisból. Az autókat úgy választjuk ki, hogy olyanokat keresünk, melyeknek nincs foglalása az adott időszakra.

Az eredményeket a felhasználói felületen listázzuk, valamint lehetővé tesszük a felhasználók számára, hogy válasszanak egy autót a foglaláshoz.

## Foglalás végrehajtása:

Ha a felhasználó elküldte a foglalási űrlapot (`submit_reservation` gomb lenyomásával), akkor az adatokat feldolgozzuk a PHP kódban. Elsőnek egy tranzakciót indítunk a biztonság érdekében. Lekérdezzük az aktuális ügyfél azonosítóját az adatbázisból, majd beszúrjuk az új ügyfelet az adatbázisba, ha még nem létezik. Ezután beszúrjuk a foglalást az adatbázisba a megfelelő adatokkal. Ha minden művelet sikeres, akkor `commit`-áljuk a tranzakciót, különben visszavonjuk (`rollback`). A felhasználót tájékoztatjuk a sikeres vagy sikertelen foglalásról.

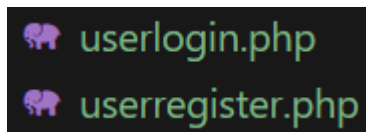
## Vue.js alkalmazás a validációhoz:

A Vue.js alkalmazás segít a felhasználói bemenetek ellenőrzésében. Ha a felhasználó nem adta meg az összes szükséges adatot, akkor figyelmeztetést kap. A Vue.js kód az űrlap validációját végzi el, valamint megakadályozza annak elküldését, ha hiányosak az adatok.

## Modal ablakok a járműleírások megjelenítéséhez:

Az elérhető autókhoz modal ablakokat rendelünk hozzá, melyekben megjelenítjük az autók leírását. Ezek a modal ablakok megjelennek, ha a felhasználó a "Description" feliratra kattint az adott autó mellett.

## Bejelentkezés és Regisztráció:



Ezek a PHP fájlok kulcsfontosságú részei a programnak, mely egy felhasználói regisztrációs és bejelentkezési rendszert valósít meg. A userregister.php fájl lehetővé teszi újabb felhasználók regisztrációját a rendszerbe, ellenőrizve az adatok megfelelőségét és biztosítva, hogy az e-mail cím egyedinek számítson. A userlogin.php fájl felhasználói bejelentkezést kezel, ellenőrizve a megadott adatokat, továbbá átirányítva a sikeres bejelentkezést követően. Ezek a fájlok direkt kapcsolatot teremtenek az adatbázissal az adatok lekérdezése, továbbá frissítése érdekében megfelelően, miközben biztosítják a felhasználók részére a biztonságos, továbbá zökkenőmentes hozzáférést a rendszerhez.

Az alábbi részekre bonthatóak:

### userlogin.php:

#### Session kezelés, ezen kívül HTML fejléc:

Az `session_start()` függvény támogatásával indítjuk a munkamenetet, ami lehetővé teszi az adatok tárolását a felhasználó részére. Az HTML fejlécbe beágyazva van a CSS stíluslapok, ezen kívül a Font Awesome ikonokhoz tartozó linkek.

#### Bejelentkezési adatok ellenőrzése:

Az űrlap elküldését a `$_POST['login']` változó ellenőrzi. Ha a felhasználó beírta az e-mail címét és jelszavát, abban az esetben azokat ellenőrzi a regusers adatbázisban tárolt adatokkal. Ha a megadott e-mail cím, továbbá jelszó egyezik a tárolt adatokkal, a felhasználót bejelentkezteti, továbbá átirányítja az index.php oldalra.

#### Bejelentkezési űrlap:

Az űrlap HTML kóddal rendelkezik, mely két input mezőt hordozhat magában az e-mail cím és a jelszó megadásához. A jelszó mezőnél egy ikon látható, melyre kattintva a jelszó megjelenítését, esetleg elrejtését lehet megváltoztatni.

userregister.php:

### Regisztrációs adatok feldolgozása:

Az űrlap elküldését az `isset($_POST['register'])` feltétel kutatja.

Amennyiben a felhasználó megadta a regisztrációhoz szükséges adatokat (felhasználónév, e-mail cím, jelszó), azokat ellenőrzi, ezen kívül feldolgozza.

Az újabb felhasználói adatokat a `regusers` táblához hozzáadja, ha azok megfelelőek.

### Regisztrációs űrlap:

Az űrlap HTML kóddal rendelkezik, mely négy input mezőt hordozhat magában a felhasználónév, e-mail cím, jelszó és jelszó megerősítése megadásához.

A jelszó mezőknél egy ikon látható, melyre kattintva a jelszó megjelenítését, esetleg elrejtését lehet megváltoztatni.