

Záródolgozat



Vakulya Viktor

Szoftverfejlesztő

5/13/A

**Kecskeméti SZC
Kandó Kálmán Technikum**

2022

Tartalomjegyzék

Témaválasztás	3
Felhasználói dokumentáció	4
Rendszerkövetelmények:	4
Setupom:	4
Programozási nyelvek – Fejlesztői eszközök	5
CSS:	5
HTML:	5
PHP:	5
SQL:.....	5
Java Script:.....	5
Visual Studio Code:.....	5
GitHUB:.....	6
PHPMYADMIN és az XAMPP:	6
Bootstrap 5.0:.....	6
REST API:.....	6
jQuery:.....	6
Ajax:.....	7
PDO:.....	7
Adatmodell	8
Adatbázisra való kapcsolódás PHP-val	10
Menü	14
Gyakori kérdések:.....	15
Autóink:	15
Kapcsolatfelvétel:	15
Bejelentkezés:	15
Foglalás	17
Kapcsolat felvétel	19
Működése:.....	19
Tesztelés	20
További fejlesztési lehetőségek	21
Források:	22

Témaválasztás

Választásomat úgy hoztam meg, hogy kisgyermekkorom óta rajongom az autókért és számomra nem is volt kérdés, hogy egy autós weblapot fogok készíteni. Így esett nekem a választás az autókölcsönzőre, ahol el tudtam képzelni milyen autókat adnék bérbe és milyen gyönyörű autók lehetnének amivel másoknak tudnék hatalmas örömet okozni.

A weboldal készítésre pedig úgy esett a választásom, hogy mindig is a honlapok készítése érdekelt a legjobban az informatika különböző ágazatain belül. Itt szabadjára tudom engedni a fantáziámat és a kreativitásomat. Szeretek alkotni és mindig jó látni, hogy itt lépésről lépésre tudok fejlődni és mindig valamilyen újdonsággal kibővíteni a weboldalam.

Ezt a témát nem csak az érdeklési köröm miatt választottam. Hanem azért is, mert egy kihívást láttam ebbe, hogy megtudjam valósítani amit megálmodtam. Több nehézség is szembejött velem. Egyik a foglaltság volt, amiről majd a 13. oldalon olvashat részletesebben. A következő a regisztráció és a belépés volt. Megoldottam azt is, hogy csak az tudjon kölcsönözni aki már regisztrált a honlapon és be is van jelentkezve.

Kiknek ajánlom a weboldalam? Egyértelműen azoknak akik szerelmesei a nagy lóerős autóknak. Megígérhetem, hogy aki autót bérel nálam egy örök emlék ként fog megmaradni az emlékekben.

Felhasználói dokumentáció

A weboldalam elkészítésére azért volt szükség, mert így egy hatalmas élményt tudok nyújtani az emberek számára.

A honlapomra látogató felhasználók a fő oldallal fog találkozni, ahol láthatja a legnépszerűbb járműveinket. Foglalni csak is az tud aki bejelentkezett, így nem kell a nevét és az e-mail címét újból beírni, ezáltal nagyon gyorsan és egyszerűen tudunk foglalni.

A weboldal teljesen reszponzív és folyamatosan teszteltem telefonos méretben is, hogy számomra és a felhasználó igényeinek is megfeleljen. Így mindig telefonos méretben teszteltem és designoltam majd csak utána, foglalkoztam a nagyképernyős mérettel is. Miért elsőnek a telefonos méret? Azért, mert többen látogatják már meg a honlapok a telefonjukon mint a számítógépükön.

A felső menü minden oldalon megjelenik. Funkciója az, hogy minden menü pontot el tud érni a felhasználó kényelmesen és egyszerűen. Telefonon jobb felül egy lenyíló lista található, amit lenyitva ugyan ezeket a menü pontokat tudjuk megtalálni.

Rendszerkövetelmények: A weboldal minden olyan felhasználónak elérhető akinek van internetkapcsolata és egy nagyon minimális rendszere. Ami az alábbiakat takarja: windows 10 abszolút hibátlanul fog fut de ennél régebbi operációs rendszereken is futni tud a honlapunk. Legalább 1 gighertz-es vagy gyorsabb processzor, 2 gigabájt RAM, legalább DirectX 9-es kompatibilitású videokártya. Ezek a specifikációk napjainkban már szerintem teljesen alap így majdnem bárki megtudja látogatni a weboldalam.

Setupom: Ahhoz, hogy megfelelően tudjuk elvégezni a munkánkat ahhoz egy megfelelő konfiguráció szükséges. A honlapon elkészítéséhez én egy AMD Ryzen 3600x processzort használtam. Videókártyám egy AMD Vega 56 8GB videokártya, ezek mellett egy 3200Mhz 16GB RAM található a gépemben. Ezzel a setuppal gond nélkül lehetett a forráskódot futtatni és így semmilyen technikai nehézségekbe nem akadtam. Ezáltal rengeteg időt spóroltam meg amit a weboldalam fejlesztésébe tudtam befektetni.

Programozási nyelvek – Fejlesztői eszközök

CSS: A CSS (Cascading Style Sheets, magyarul: lépcsőzetes stíluslapok) a számítástechnikában egy **stílusleíró nyelv**, mely a HTML vagy XHTML típusú strukturált dokumentumok megjelenését írja le. Ezenkívül használható bármilyen XML alapú dokumentum stílusának leírására is, mint például az SVG, XUL. A CSS-t a weblapok szerkesztői és olvasói egyaránt használhatják, hogy átállítsák vele a lapok színét, betűtípusait, elrendezését, és más megjelenéshez kapcsolódó elemeit. A tervezése során a legfontosabb szempont az volt, hogy elkülönítsék a dokumentumok struktúráját (melyet HTML vagy egy hasonló leíró nyelvben lehet megadni) a dokumentum megjelenésétől (melyet CSS-sel lehet megadni). Az ilyen elkülönítésnek több haszna is van, egyrészt növeli a weblapok használhatóságát, rugalmasságát és a megjelenés kezelhetőségét, másrészt csökkenti a dokumentum tartalmi struktúrájának komplexitását.

HTML: A HTML(angolul: HyperText Markup Language=hiperszöveges jelölőnyelv) egy leíró nyelv, melyet weboldalak készítéséhez fejlesztettek ki, és mára már internetes szabvánnyá vált a W3C (World Wide Web Consortium) támogatásával. HTML általában szöveges állományokban található meg olyan számítógépeken, melyek az internethez kapcsolódnak. Ezek az állományok tartalmazzák azokat a szimbólumokat, amelyek a megjelenítő programnak leírják, hogyan is kell megjeleníteni illetve feldolgozni az adott állomány tartalmát. Leggyakoribb megjelenítő programok a webböngészők.

PHP: A PHP egy általános szerveroldali **szkriptnyelv** dinamikus weblapok készítésére. Az első szkriptnyelvek egyike, amely külső fájl használata helyett HTML oldalba ágyazható. A kódot a webszerver PHP feldolgozómodulja értelmezi, ezzel dinamikus weboldalakat hozva létre.

SQL: Az SQL, azaz Structured Query Language(strukturált lekérdezőnyelv) relációsadatbázis-kezelők lekérdezési nyelve.

JavaScript: A JavaScript programozási nyelv egy **objektumorientált**, prototípus-alapú szkriptnyelv, amelyet weboldalakon elterjedten használnak. A javascript kód vagy a html vagy egy külön fájlban található(.js).

Visual Studio Code: Bármilyen szövegszerkesztővel tudjuk szerkeszteni a weboldalunkat, de szerintem ez a legjobb **ingyenes** szerkesztő, mert megtudja nekünk írni a weboldalunk alapját a shift plusz négyes gomb lenyomásával ami után már csak egy enter kell ütnünk. Ha lekezdünk begépelni betűket akkor felhossa, hogy milyen tagek vagy parancsok közül tudunk választani. Megírja nekünk a nyitó és bezáró tageket. Bárki aki megkérdezte tőlem, hogy milyen szerkesztőt ajánlok azoknak csak a Visual Studio Codeot ajánlottam.

GitHub: Ebben a tanévben fedeztem fel a githubot ami nagyon megkönnyítette az életemet az adat hordozás terén, mert a regisztrálást követően ingyenes repókat tudunk létrehozni és oda feltudjuk tölteni a kész munkáinkat. Lehet private vagy publikusra is rakni a feltöltött munkáinkat. Amiért még nagyon szerettem az az volt, hogy kompatibilis az előbb megemlített Visual Studio Coddal. Ami annyit takar, hogy ha összekapcsoljuk a két programot akkor utána pár gomb megnyomásával feltudjuk tölteni a frissen módosított munkánkat. Így a munkánkat nagy biztonsággal tudhatjuk és nem kell attól félnünk, hogy elhagynánk.

PHPMYADMIN és az XAMPP: Egy webes felületű kliens az XAMPP-on belül. Adatbázisok kezelésére szolgál.

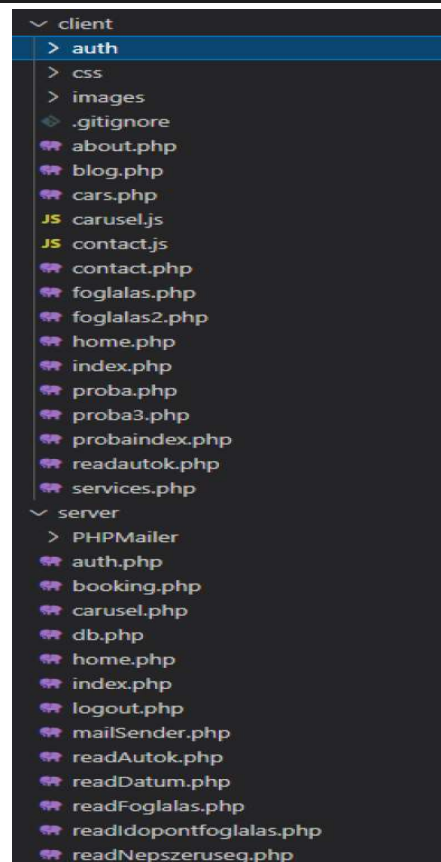
Bootstrap 5.0: A bootstrap 5.0 verzióját használtam. Azért érdemes használni, mert előre megírt osztályok vannak és így bonyolult formázásokat csak egy paranccsal megtudunk oldani. Legnagyobb előnye az, hogy könnyedén teljesen reszponzív tudjuk varázsolni a weboldalunkat.

```
<div class="container">
  <div class="row border border-dark fo">
    <div class="col-lg-12 text-center">
```

Ez a kód részlet tökéletesen szemlélteti, hogyan kell elképzelni a bootstrap működését. pl.: border – keretet állít be körbe, border-dark – sötét keretet ad a div-nek.

REST API: Egy állapotmentes (stateless) szoftver architektúra megoldás, ahol a meglévő webes standardokat használjuk ki az alkalmazás-alkalmazás **kommunikációban** a szerveren vagy kliensen tárolt objektumok állapotának átvitelére. Tipikusan a HTTP szabványban található szabályokat követi: HTTP státuszkódok, HTTP igék és HTTP fejléc kezelés.

jQuery: A jQuery népszerű JavaScript könyvtár, mely a HTML kód és a kliensoldali JavaScript közötti kapcsolatot hangsúlyozza.



Ajax:

Az Ajax (Asynchronous JavaScript and XML) interaktív webalkalmazások létrehozására szolgáló webfejlesztési technika. A weblap kis mennyiségű adatot cserél a szerverrel a háttérben, így a lapot nem kell újratölteni minden egyes alkalommal, amikor a felhasználó módosít valamit. Ez növeli a honlap interaktivitását, sebességét és használhatóságát.

Az Ajax a következő technikák kombinációja:

- XHTML (vagy HTML) és CSS a tartalom leírására és formázására.
- DOM kliensoldali szkriptnyelvekkel kezelve a dinamikus megjelenítés és a már megjelenített információ együttműködésének kialakítására
- XMLHttpRequest objektum az adatok aszinkron kezelésére a kliens és a webszerver között. Néhány Ajax keretrendszer esetén és bizonyos helyzetekben IFrame-et használnak XMLHttpRequest objektum helyett.
- XML formátumot használnak legtöbbször az adattovábbításra a kliens és a szerver között, bár más formátumok is megfelelnek a célnak, mint a formázott HTML vagy a sima szöveg.

Előnyei: Az Ajax alapú weboldalak úgy viselkednek, mint a különálló programok. Amikor a felhasználó egy közönséges oldalon kattint egy linkre, a teljes oldal újratöltése egy elég nehézkes folyamat. Ehelyett az Ajax által dinamikusan frissített oldalak jóval gyorsabban reagálnak a felhasználó kérésére.

Hátránya: A dinamikusan létrehozott oldalak nem regisztrálják magukat rendesen a böngésző előzményei közé, így például a Vissza gomb sem működik megfelelően.

PDO:

Egy natív absztrakciós réteg a webfejlesztés során. Használatával egységes formában kezelhetünk többféle adatbázist, és jobb esetben minimális módosítással cserélhetjük ki az adatbázis kiszolgáltót. Előnye, hogy a PHP része, illetve mivel C alapú, így elég gyors is. A PDO illeszkedik a PHP OOP irányú vonalához, szóval egy osztályként kapjuk kézhez.

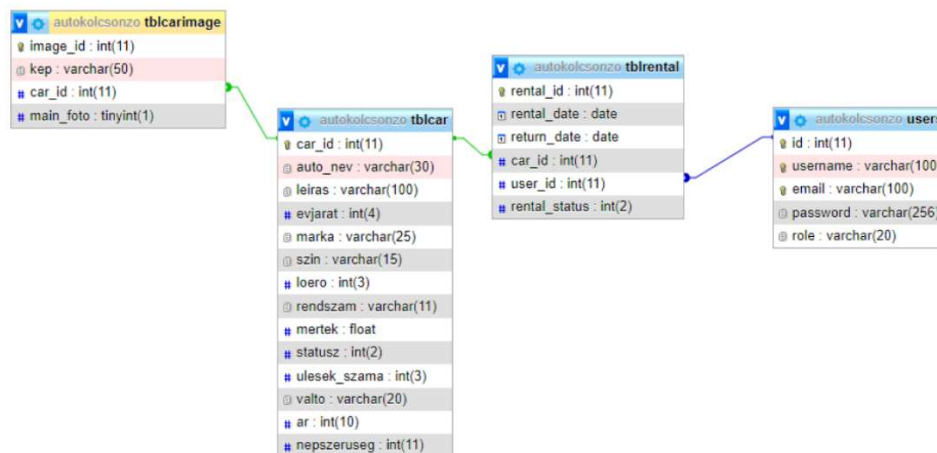
Az egyik nagy előnye a PDO-nak, hogy képes a parancsokba behelyettesíteni paramétereket úgy, hogy az adatok escapelését is elvégzi.

Ennek az SQL injection miatt van szerepe, ahol olyan adatot küldhetnek be amit előszűrés nélkül SQL parancsnak értelmezhet a rendszer."



Adatmodell

Az általunk kigondolt weboldal mellé elengedhetetlen az hogy legyen egy jól felépített adatbázisunk. A PHPMYADMIN azért is jó választás az adatbázisunk elkészítéséhez, mert grafikus felületű és felhasználó barát. Plusz extra benne az, hogy teljesen magyar a program.



tblcar: ebben a táblában az autóknek az adatai vannak eltárolva. Car_id ennek a táblának a kulcsa ezzel tudok más táblázatokban az autókra hivatkozni. Auto_nev itt az autók neve található. Leiras pont csak egy kis ismertetőt takar az autóról. A többi táblában az autó specifikációi vannak eltárolva.

tblcarimage: itt az autók képeinek a neve található meg. A képeket a projecten belüli images mappában található. Ez a táblázat össze kapcsolódik a fő tblcar táblánkkal. A képeket azért tároltam el külön táblában, mert így egy autóhoz akár több képet is lehet rendelni.

tblrental: foglalások találhatóak itt meg. Van egy kezdeti(rental_date) és egy vég dátum(return_date). Van egy user_id is aminek nagyon fontos szerepe van mivel így tudjuk, hogy melyik felhasználó melyik járművet és mennyi időre bérelte ki. A másik fontos mező még a car_id így tudhatjuk meg melyik autót foglalta le.

users: itt a felhasználók adatai találhatóak. A bejegisztrált felhasználónév és jelszó található. Pluszba még a felhasználó jogosultsága(pl:admin vagy user). A jelszavakat az adatbázisban csak titkosítva vannak eltárolva így nem kell félnie a felhasználóknak, hogy innen az adataikat eltudják lopni.

Titkosítás módszere: a PHP kínál számunkra egy megfelelően biztonságos hash függvényt a titkosításra. Ez pedig nem más, mint a "password_hash()". Ez a függvény azért nagyon biztonságos, mert egy nagyon erős hasítóalgoritmust (hash) használ, illetve a random salt módszernek hála megelőzhető a szivárványtábla és a szótár alapú támadás.

```
function register($db){  
    extract($_POST);  
    $pw=password_hash($password,PASSWORD_DEFAULT);  
    $sql="insert into users (username,email,password) values (?,?,:)";  
    $stmt=$db->prepare($sql);  
    $stmt->execute([$username,$email,$pw]);  
    if($stmt)  
        echo true;  
    else  
        echo false;  
}
```

Megvalósítás: Ennek az adatbázisnak a kitalálása és megtervezése rengeteg idő kellett. Többször is újra kellett kezdenem, mert mindig rájöttem a hibáimra és a hiányosságaimra. Úgy gondolom, hogy egy teljesen komplex és összetett adatbázist sikerült létrehoznom. A dolgomat az is egyszerűbbé tette, hogy előtte pár évig foglalkoztunk adatbázisok kezelésével és létrehozásával.

Adatbázisra való kapcsolódás PHP-val

A PHP azért jó, mert könnyedén lehet csatlakozni vele az adatbázisunkra.

```
<?php
$host = 'localhost';
$db_name = 'autokolcsonzo';
$db_username = 'root';
$db_password = '';
$options = [PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION, PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC];
try{
    $db = new PDO("mysql:host=$host;dbname=$db_name; charset=utf8",$db_username,$db_password,$options);
}catch(PDOException $e) {
    echo "hiba: az adatbazis kapcsolodas sikertelen !!!". $e->getMessage();
    exit;
}
?>
```

A \$host-ban található a helyi kiszolgáló, ha az XAMPP-ot nem indítottuk le akkor sajnos nem lesz működőképes a weboldalunk és nem fog betölteni.

\$db_name azt jelenti, hogy mi az adatbázisunk neve és hogy erre az adatbázisra kell csatlakoznia.

\$db_username ebbe ha azt írjuk, hogy root akkor az azt fogja jelenteni, hogy a mappából kell elindítani a programot.

\$db_password ide csak akkor kell beírni bármit is ha az adatbázist csak a megfelelő jelszóval tudunk rá csatlakozni.

```
<?php
require_once 'db.php';
$sql="SELECT * FROM `tblcar` inner JOIN tblcarimage ON tblcar.car_id=tblcarimage.car_id;";
$stmt=$db->query($sql);
echo json_encode($stmt->fetchAll());
?>
```

Bármilyen SQL-es utasítást betudunk írni, majd azt utána megjeleníteni a webes felületen. A lekérdezésünket ajánlott elsőnek a PHPMYADMIN-ban lefuttatni, mert ha ott jól működik akkor a weboldalunkon is működni fog, ezzel a hiba lehetőségek számát tudjuk csökkenteni.

A fentiek alapján tudtunk server oldalon csatlakozni az adatbázishoz. Majd utána sikerült megírunk a lekérdezésünket. De még sajnos nem fog működni nekünk, mert kliens oldalon is meg kell írunk a kódunkat.

Ez a kódrészlet tökéletesen szemlélteti, hogyan is tudjuk elérni a korábban server oldalon megírt lekérdezésünket.

```
<script>
  fetch('../server/readAutok.php')
    .then(response=>response.json())
    .then(data=>szakrender(data))

  function szakrender(data){
    console.log(data);
    let tblStrszak = '';

    for(let obj of data) {
      tblStrszak += `
      <div class="col-lg-3 mx-2 col-md-6 mb-3 border border-secondary bg-secondary rounded pt-5 pb-5 ps-3 ">
        <div class="item-1 ">
          <div class="kepek">
            
          </div>
          <div class="item-1-contents">
            <div class="text-center">
              <h3><p>${obj.auto_nev}</p></h3>
              <div class="rating">
                ${csillag(obj.mertek)}
              </div>
              <div class="rent-price"><span><h3>${obj.ar}HUF/nap</h3></span></div>
            </div>

            <div class="d-flex action">
              <?php
                | if(isset($_SESSION['username'])):
              >
              <a href="index.php?program=foglalas.php&id=${obj.car_id}" class="mt-3 btn btn-primary m-auto">Foglalás</a>
              <?php
                | else:
              >
                <p>Jelentkez be a kölcsönzéshez</p>
              <?php
                | endif
              >
            </div>
          </div>
        </div>
      </div>`
    }
  }
```

Fetch: ide kell beírni annak a fájlnek a nevét amiben megtalálható a lekérdezésünk. Miért van ott a ../server/? Azért, mert a fájlmag egy server mappában tároltam el a fő mappámon belül. Így lehet megadni az útvonalat, hogy megtalálja. Ha nem adtam volna meg neki az útvonalat akkor nem találta volna meg és nem működött volna.

Csinálunk egy functiont aminek adhatunk bármilyen nevet pluszba lesz egy paramétere, jelen esetben nekem a data. A data-ba fognak bekerülni a lekérdezéseinknek az adatai és a data egy tömb lesz.

Így találhatók meg benne az adatok:

```
▼ Array(6)
▶ 0: {car_id: '2', auto_nev: 'BMW M4', leiras: 'Jó állapotban lévő.', evjarat: '2020', marka: 'BMW', ...}
▶ 1: {car_id: '3', auto_nev: 'RS 6', leiras: 'Jó', evjarat: '2021', marka: 'Audi', ...}
▶ 2: {car_id: '4', auto_nev: '3200', leiras: 'Megkímélt, frissen szervizelt.', evjarat: '2006', marka: 'BMW', ...}
▶ 3: {car_id: '5', auto_nev: 'Giulia', leiras: 'Új nyári gumikkal, 50 km ez előtt cserélt motor olaj.', evjarat: '2021', marka: 'Alfa Romeo', ...}
▶ 4: {car_id: '6', auto_nev: 'AMG GT', leiras: 'Brutális teljesítmény, a száguldás szerelmeseinek.', evjarat: '2022', marka: 'Mercedes', ...}
▶ 5: {car_id: '7', auto_nev: 'Camaro', leiras: 'Jó', evjarat: '2015', marka: 'Chevrolet', ...}
length: 6
```

Létrehozunk egy változót bármilyen néven(let példa) majd utána **for** ciklus segítségével végig megyünk a data tömbünkön. De miért is jó ez nekünk? Azért, mert ha nem használnánk for akkor csak az első autó adata jelenne meg. For ugyebár arra való, hogy egy tömbnek az első elemétől az utolsó eleméig végig nézze.

Ezt követően a létrehozott változónkba beletöltjük az adatokat. Annyiszor fog lefutni az adat betöltés ahány adat szerepel a tömbben.

A változóba „+=" töltjük be az elemeket, mert ha csak egyenlőséget raknánk oda akkor csak egy adat maradna benne. A kód úgy tűnhet elsőre, hogy sima macska körmök között helyezkedik el, de pedig azok backtick(Alt Gr+7 a kombinációja). Ez után már csak annyi a dolgunk, hogy ide ahogy elképzeltük a designt úgy megírjuk mint egy normális HTML kód rész lenne.







```
<div class="site-section hatter">
  <div class="container">
    <div class="row justify-content-center " id="sorika">

    </div>
  </div>
</div>
```

```
document.getElementById('sorika').innerHTML = tblStrszak;
```

Az adatokra obj. amit szeretnénk adatot írunk oda. Az obj. utáni nevek az adatbázisunk mezőinek nevei ezért is érdemes úgy elnevezni őket, hogy könnyen értelmezhetőek. Ha ezzel elkészültünk akkor boldogan lefrissítjük az oldalunk és várjuk, hogy megjelenjenek az adatok. Hiába frissítünk nem jelenik meg. De miért is? Azért, mert nincs amibe megjelenítse az adatainkat. Ezért egy div-et hozunk létre aminek egy tetszőleges id adunk és abba bele rakjuk az adatokat. Ezt úgy kell elképzelni, hogy csinálunk egy div-et aminek a belseje üres, mert azt a kódrészletet amit az előbb megírtunk azt abba div-ünkbe bele rakjuk. Ezt a weboldalak úgy érzékelik mintha a kód ott lenne és megjeleníti nekünk.

Eredmény:

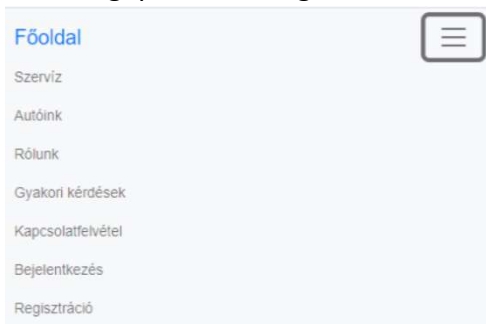
 <p>BMW M4 40000 HUF/nap</p> <p>Foglalás</p>	 <p>RS 6 30000 HUF/nap</p> <p>Foglalás</p>	 <p>320D 20000 HUF/nap</p> <p>Foglalás</p>
 <p>Giulia 30000 HUF/nap</p> <p>Foglalás</p>	 <p>AMG GT 60000 HUF/nap</p> <p>Foglalás</p>	 <p>Camaro 40000 HUF/nap</p> <p>Foglalás</p>

Az hogy hogyan jelenjenek meg nekünk az adatok az már csak a mi elképzelésünk része.

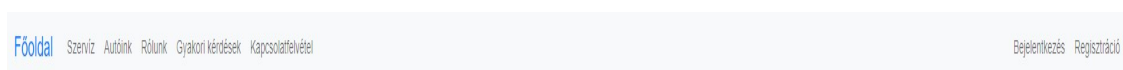
Itt is alkalmaztam bootstrapt, hogy kisebb méreteknél ne három darab autó legyen egymás mellett hanem csak egy.

Menü

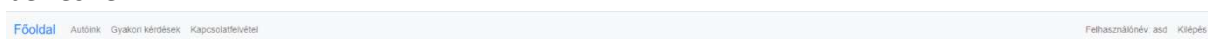
A fenti menü a weboldalam összes oldalán megtalálható. Mivel ha bármelyik menüpontra rákattintunk akkor gyorsan tudunk oldalakat váltani. A menüm teljesen a bootstrappel lett elkészítve, ha telefonon nézik a menüt akkor teljesen más mint aki számítógépen nézi meg.




Így mindenkinek egyszerű lesz a menü kezelése. Egy egyszerű és nem túl bonyolított menüt készítettem.



Főoldal: Itt a legnépszerűbb autók találhatók meg és néhány fontos információ bérléshez.



Vitya autókölcsönzője




BMW M4

40000HUF/nap

- Márka BMW
- Típus BMW M4
- Váltó Automata
- Évjárat 2020
- Teljesítmény 500 LE
- Ülések száma 5

Foglalás




320D

20000HUF/nap

- Márka BMW
- Típus 320D
- Váltó Manuális
- Évjárat 2006
- Teljesítmény 163 LE
- Ülések száma 5

Foglalás



AMG GT

60000HUF/nap

- Márka Mercedes
- Típus AMG GT
- Váltó Automata
- Évjárat 2022
- Teljesítmény 800 LE
- Ülések száma 2

Foglalás

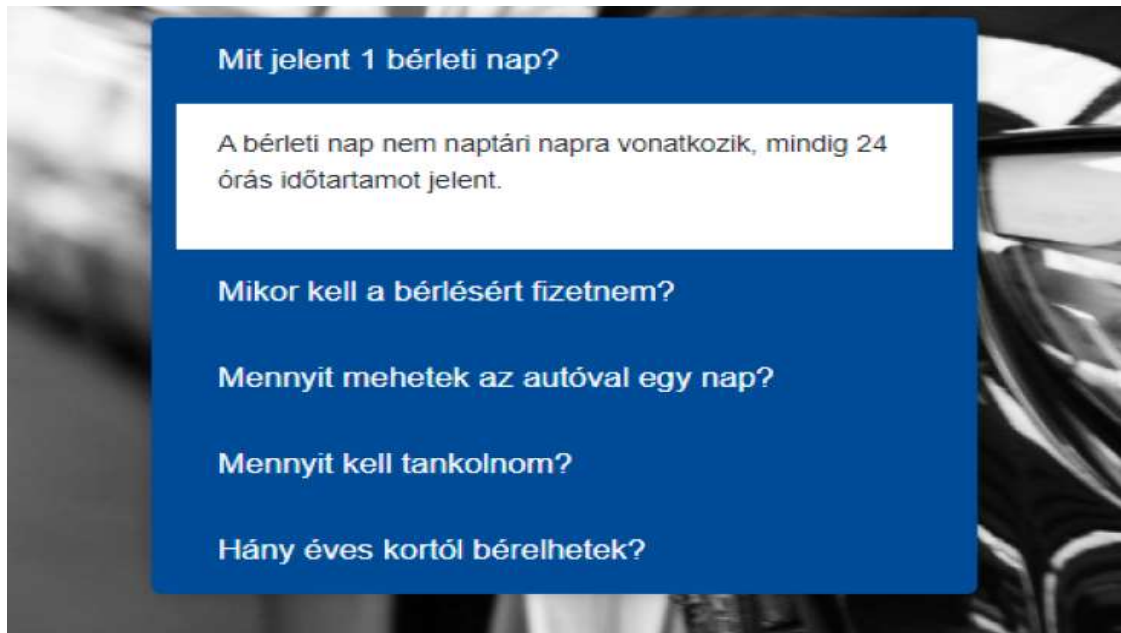
Amik az árba beletartoznak

- Biztosítási díjat
- Ha rendelésbe az autó azonnali segély nyújtást biztosítunk és csere autót.
- 500km/nap, ezt meghaladón +6000km.

Bérlési díj nem tartalmazza

- A szabályszerűségeket követő büntetéseket.
- Autópálya és parkolási díjakat.
- Üzemanyag díjat

Gyakori kérdések: A legfontosabb kérdésekről olvashattok. Aki erre a pár kérdésre a választ végig olvassa az garantáltan minden részletet tudni a bérletről.



Autóink: Az összes autónk megtalálható itt. Itt az autóról egy darab képet lehet megtekinteni és néhány adatát. Ha valaki bővebb információt szeretne megtudni a járművekről az nyomjon rá a foglalás gombra.

Kapcsolatfelvétel: A felhasználók nekem tudnak majd e-mailt küldeni. Így ha bármilyen kérdés merül fel bennük akkor arra pár órán belül fogok tudni válaszolni nekik. A választ majd a megadott e-mail címükre várják.

Bejelentkezés: A felhasználó ha már regisztrált akkor be fog tudni jelentkezni. Miért fontos a bejelentkezés a felhasználók számára? Azért, mert ha nincs bejelentkezve akkor nem fog tudni autót foglalni. Ez úgy lett megoldva a weboldalon, hogy amíg valaki nincsen bejelentkezve akkor annak a foglalás gomb nem fog megjelenni. Ha be vagyunk jelentkezve akkor a bejelentkezés és a regisztráció menü pontok elfognak

Felhasználónév: Vitya Kilépés

tűnni és a felhasználó nevünk és a kilépés menü pontok fognak megjelenni.

A foglalás gomb, hogy ne jelenjen meg annak aki nem regisztrált be úgy működik, hogy amikor bejelentkezik valaki akkor a SESSION tömbbe eltároljuk a felhasználó nevét.

Majd a gomboknál egy PHP blokkon belül megvizsgáljuk, hogy a SESSION tömbünkbe van-e username ha nincs akkor a foglalás gomb helyett a „Jelentkezzen be a kölcsönzéshez szöveg fog megjelenni”.

```
<?php
    if(isset($_SESSION['username'])):
?>
<a href="index.php?program=foglalas.php&id=${obj.car_id}" class="mt-3 btn btn-primary m-auto">Foglalás</a>
<?php
    else:
?>
    <p>Jelentkez be a kölcsönzéshez</p>
<?php
    endif
?>
```

Regisztráció: A szokásos felhasználónév, e-mail és jelszó kell csak. A regisztráció fontosságáról már az előbbi pontban tájékozódott.

Itt is azonnal beíródik az adatbázisba az adatok és hogy milyen vizsgálatokat végez még a regisztrációs felület arról a tesztelés pontba olvashat részletesebben.

					id	username	email	password	role		
<input type="checkbox"/>		Módosítás		Másolás		Törölés	1	asd	asd	\$2y\$10\$xBgnCcVXasQU2h9zPG3TO0taFGbpAZg45bXXVW0MX5N...	user
<input type="checkbox"/>		Módosítás		Másolás		Törölés	3	Vitya	teszt@gmail.com	\$2y\$10\$YfYHJIhyS52CEp0yAVzak.EtfapU93lIFZdmDIE9XIb...	user
<input type="checkbox"/>		Módosítás		Másolás		Törölés	5	adminjuguiqwgru3	dawnd	\$2y\$10\$fzxvOT73TLyoAYGdSZaYze6Z1pCuOn.y96ZAhtydKFd...	user

Foglalás

A foglalás megoldása egy bonyolultabb mechanikája már a weboldalnak, hiszen ha rákattint a felhasználó egy autóra, hogy le szeretné foglalni akkor a honlapnak tudnia kell, hogy melyik autó adataira lesz szükség a továbbiakban.

A megvalósításhoz mire lesz szükségünk? Nem másra mint egy SQL lekérdezésre.

```
<?php
require_once 'db.php';
$sql="SELECT * FROM `tblcar` inner JOIN tblcarimage ON tblcar.car_id=tblcarimage.car_id
WHERE tblcar.car_id={$_GET['id']}";
$stmt=$db->query($sql);
echo json_encode($stmt->fetch());
?>
```

```
<a href="index.php?program=foglalas.php&id=${obj.car_id}" class="mt-3 btn btn-primary m-auto">Foglalás</a>
```

Ebben a lekérdezésben az autók táblából az autók összes adatát elkérjük de viszont feltételbe azt írjuk, hogy csak a megadott id autónak az adatait szeretnénk elkérni. Miután elkértük az id megvan az adatok is. A következő teendők átírányítani a felhasználót egy másik oldalra ahol már az általa kiválasztott autók adatai láthatók.

Itt több információ megtalálható az általa kiválasztott járművekről. Meg még látható lesz egy kezdeti dátum és egy vég dátum. Itt ha rákattint bármelyik dátumra akkor látni fogja, hogy melyik napokon van már lefoglalva az autó. Fontos csak akkor tudunk érvényesen foglalni ha a vég dátum későbbre esik mint a kezdeti dátum és nem lehet a maitól számítva korábbi napokra foglalni. Ha valaki még is megpróbálná akkor amikor rányom a foglalás gombra egy rossz dátum felirat fog neki megjelenni.

A(z) localhost közlendője

3 nap van a két dátum között.

→Ár:120000 Ft.

OK

- 2020
- Király kék
- 500



















Kezdeti dátum: 01-05-2022

Vég dátum: 04-05-2022

Ha megfelelő időpontokat kiválasztottuk és a fenti követelményeknek is megfelel akkor amint rányomunk a foglalás gombra akkor azonnal bekerül az adatbázisba a foglalás időpontja. Látszodni fog az adatbázisban, hogy melyik felhasználó, mettől, meddig és hogy melyik autót bérelte ki.

A sikeres bérlést követően felugrik a képernyőn egy ablak ami kiírja, hogy sikerült a foglalás és hogy mennyi pénzbe fog kerülni.

Majd utána ha ráfrissít újra a felhasználó akkor már azonnal megjelenik, hogy arra az időpontra már nem tudnak foglalni. Így egy nagyon gyors és jól működő rendszert kapunk, amivel egyszerűen tudjuk a foglalásokat kezelni.

<div><div>←T→</div><div></div></div>							rental_id	rental_date	return_date	car_id	user_id	rental_status
<div><div><input type="checkbox"/></div><div>Módosítás</div></div>	<div><div>Másolás</div></div>	<div><div>Törlés</div></div>	1	2022-01-28	2022-01-31	2	3	0				
<div><div><input type="checkbox"/></div><div>Módosítás</div></div>	<div><div>Másolás</div></div>	<div><div>Törlés</div></div>	2	2022-04-10	2022-04-15	2	1	0				
<div><div><input type="checkbox"/></div><div>Módosítás</div></div>	<div><div>Másolás</div></div>	<div><div>Törlés</div></div>	3	2022-09-04	2022-09-08	2	3	0				
<div><div><input type="checkbox"/></div><div>Módosítás</div></div>	<div><div>Másolás</div></div>	<div><div>Törlés</div></div>	4	2022-04-12	2022-04-15	2	1	1				
<div><div><input type="checkbox"/></div><div>Módosítás</div></div>	<div><div>Másolás</div></div>	<div><div>Törlés</div></div>	26	2022-03-10	2022-01-01	2	3	0				
<div><div><input type="checkbox"/></div><div>Módosítás</div></div>	<div><div>Másolás</div></div>	<div><div>Törlés</div></div>	37	2022-03-25	2022-03-27	2	1	0				

Kezdeti dátum:

Vég dátum:

◀

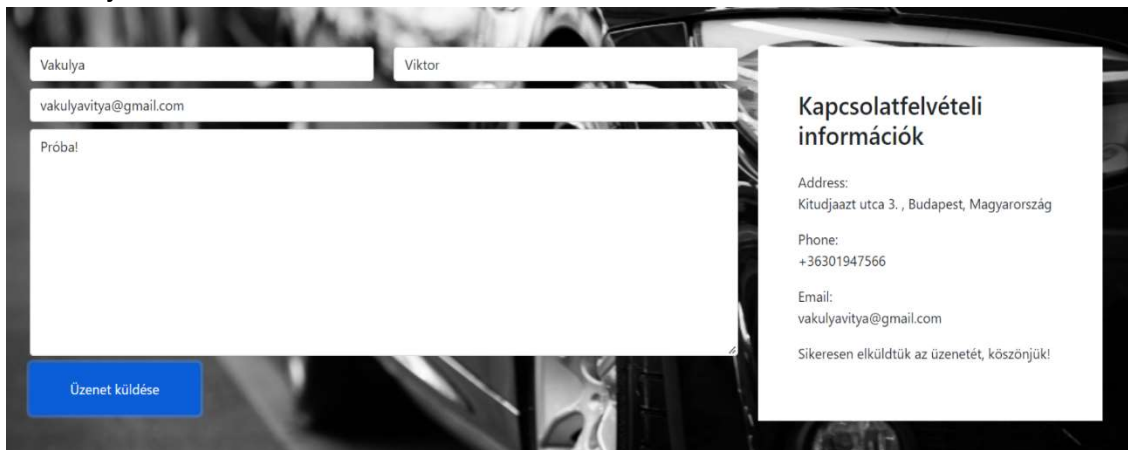
April 2022

▶

Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Kapcsolat felvétel

A kapcsolatfelvétel egy nagyon fontos része a weboldalnak, hiszen így tud a felhasználó velünk kapcsolatot felvenni. Egy család és egy keresztnév, egy e-mailt és a közlendőjét kell leírni. Nagyon fontos, hogy jól töltsse ki a felhasználó a mezőket, ha nem jól tölti ki akkor nem fogok tudni válaszolni neki. Az e-mail címre kell a legjobban oda figyelni. Ha az nem jó akkor a válaszadás lehetetlen.



The image shows a contact form on the left and a confirmation message on the right. The form has fields for 'Család' (Family) with the value 'Vakulya', 'Keresztnév' (Surname) with 'Viktor', 'Email cím' (Email address) with 'vakulyavitya@gmail.com', and a 'Közlendő' (Message) field with 'Próba!'. A blue 'Üzenet küldése' (Send message) button is at the bottom. The confirmation message on the right, titled 'Kapcsolatfelvételi információk', lists the submitted details: Address (Kitudjaazt utca 3., Budapest, Magyarország), Phone (+36301947566), and Email (vakulyavitya@gmail.com), ending with 'Sikeresen elküldtük az üzenetét, köszönjük!' (We successfully sent your message, thank you!).

Működése: Elsőnek is a mezők értékeit el kell tárolnunk egy tömbben, az input elemeken egy id és egy name kell amiknek meg kell egyezniük.. Ha ez meg van akkor a PHP segítségével tovább küldjük. Majd utána egy \$mail-ben beállítjuk, hogy melyik proton hogyan és melyik e-mailről küldje el az üzenetet. Levél küldéshez egy külön e-mailt kellett készítenem, mert a jelszavát is meg kell adni a programnak ahhoz, hogy működni tudjon, így nem kell a felhasználónak bejelentkeznie, ezért fontos, hogy pontosan adja meg az e-mail címét. Ha ez meg van akkor meg kell adnunk, hogy melyik címre küldjük és be kell tölteni még az adatokat a változóba amit a \$_POST segítségével érünk el. De az adatokat már \$content-be küldjük. Csak akkor fog működni a küldés, hogy amiről küldjük az üzenetet azon a fiókon be kell kapcsolni az ismeretlen erőforrásokat így tud belépni és elküldeni az e-mailt.



zarodolgozat123@gmail.com

címzett: én ▾

Családnév: Vakulya

Keresztnév: Viktor

Email cím: vakulyavitya@gmail.com

Üzenet: Próba!

Tesztelés

A weboldalamat több féle tesztelésnek vetettem alá.

Az egyik nagyobb tesztelés a regisztráció volt. Pl.: ki írja ha valaki regisztrálni szeretne valamilyen felhasználó néven de az már foglalt akkor azt a weboldal ki írja neki, kiírja ha az adott e-mail címen már regisztráltak vagy ha a kettő jelszó nem egyezik meg.

Sikertelen bejelentkezés!

Bejelentkezés

Felhasználónév

Email

Jelszó

☐ Jelszó megjegyzése

Bejelentkezés

A következő tesztelés a bejelentkezésnél történt, hogy a felhasználó által már létrehozott felhasználó neve mellett hogy jó jelszót ír-e be.

A foglalásnál is teszteltem, hogy megfelelő adatok jelennek meg.

Az utolsó tesztelés a végleges foglalásnál történt, hogy a kiválasztott dátumok a fentiekben olvasott kirtériumainak megfelelnek-e, ez jól működött. Utána még azt kellett tesztelnem, hogy megfelelően számolja-e ki a lefoglalt napok számát és ehhez megfelelően számolja ki a foglalás árát.

Foglalt felhasználónév

Regisztráció

Felhasználónév

További fejlesztési lehetőségek

Több további fejlesztési ötletem is van amit majd a közeljövőben szeretnék megvalósítani.

Elsőnek is azt szeretném megoldani majd, hogy ha egy felhasználó egy autót ki bérelt de még is meggondolta magát akkor még a lefoglalás előtt egy héttel letudja mondani. Így nem lennének felesleges kibérelve az autók, ezáltal biztosabb lenne a bevétel és nem tudunk akkor így biztos felhasználókat veszíteni. Mi történik aki egy héten belül szeretné lemondani a foglalását? Ebben az esetben ahány napra foglalta le annyiszor 5000ft kell fizetnie ez azért szükséges, mert így a felhasználók biztosan átgondolják azt, hogy ne foglalják le az autóinkat csak poénból.

A másik új funkció az lenne, hogy ha kibérelte az autót majd utána vissza hozta akkor tudjon minket értékelni. Ennek az lenne a célja, hogy ha egy új felhasználó rátalálna a honlapomra akkor látja, hogy milyen elégedettek voltak az emberek. Ezzel tudnám növelni a bérletek számát és népszerűsíteni a honlapomat.

A másik nagy fejlesztés még az lenne, hogy egy admin felületet hoznék létre a weboldalon ami csak az admin jogosultságú felhasználónak jelenne meg a bejelentkezést követően. Erre azért lenne nagy szükség, mert akkor nem a PHPMYADMIN-on belül kéne felvinni az adatokat hanem grafikusán a weboldalon keresztül tudnánk az összes adatot kezelni. Ennek több nagyon fontos előnye van nem csak olyan személy tudna adatokat felvinni aki ért is az adatbázisokhoz, így bárki feltudja vinni az adatokat.

Források:

[https://hu.wikipedia.org/wiki/Ajax_\(programoz%C3%A1s\)](https://hu.wikipedia.org/wiki/Ajax_(programoz%C3%A1s))

<https://hu.wikipedia.org/wiki/JQuery>

https://hu.wikipedia.org/wiki/Cascading_Style_Sheets

<https://hu.wikipedia.org/wiki/JavaScript>

<https://getbootstrap.com/docs/5.0/getting-started/introduction/>

<https://stackoverflow.com/>

<https://github.com/>

<https://pixabay.com/photos/>

<https://www.pexels.com/hu-hu/>

Zárodolgozatom GitHubos linkje:

<https://github.com/Vitya0908/VakulyaViktorZarodolgozat.git>