

**Debreceni Egyetem**  
**Informatika Kar**

## **EGY AUTÓKÖLCSÖNZŐ WEBOLDALÁNAK FEJLESZTÉSE**

Témavezető:

Dr. Kuki Attila

egyetemi adjunktus

Készítette:

Marincsák Gergely

programtervező informatikus

**Debrecen**

**2009.**

*Köszönetemet fejezem ki elsősorban Dr. Kuki Attila egyetemi adjunktus úrnak, hogy felügyelte és tanácsaival elősegítette dolgozatom elkészítését, továbbá a Debreceni Egyetem Informatikai Kar tanárainak, hogy az évek során tapasztalt elkötelezett munkájukkal hozzásegítettek az informatikai tudásom megszerzéséhez.*

# Tartalomjegyzék

I. Bevezetés .....	4.
II. Az oldal felépítése .....	7.
III. Megvalósítás .....	11.
Program összes funkciójának betöltése .....	11.
Csatlakozás az adatbázishoz .....	11.
SqlSelect és SqlMod .....	12.
beSession() .....	13.
Oldal betöltése .....	15.
Bejelentkezés .....	15.
Kijelentkezés .....	16.
Profil .....	16.
Regisztráció .....	18.
Gépjárművek listázása .....	20.
Gépjárművek kölcsönzése .....	22.
Menük .....	25.
Kezdőlap, hírek, elérhetőség .....	26.
Beállítások .....	27.
FCKEditor .....	28.
Új gépjármű hozzáadása .....	29.
Gépjárművek adatainak szerkesztése .....	30.
Kölcsönzések listázása .....	33.
Felhasználók listázása .....	35.
Kinézet .....	35.
IV. Összefoglalás .....	37.
V. Felhasznált technológiák, alkalmazások .....	38.
VI. Irodalomjegyzék .....	38.
VII. Függelék .....	38.

## I. Bevezetés

Mint ahogy az a címből kiderül, a szakdolgozatomban egy autókölcsönző weboldalának az elkészítéséről, annak menetéről, fázisairól írok a hozzá elkészített példa alapján, melyet CD-n mellékletként csatoltam.

De még mielőtt belevágnék a kifejtésbe, had essen pár szó a PHP-ről, ugyanis ezt a technológiát választottam a weboldal alapjául. Amikor az 5. félévben leadtam a szakdolgozat téma választásomat, akkor még semmi konkrétat nem tudtam ezen technológiáról, csak annyit, hogy meg szeretném ismerni. Ezért aztán még ugyanezen félévben megismerkedtem vele a Webfejlesztés PHP-vel szabadon választható óra keretében, de alaposabb ismereteket csak a mostani dolgozat elkészítési ideje alatt szereztem.

Na de mi is az a PHP? Hypertext Preprocessor, nyílt forráskódú, számítógépes szkriptnyelv, legfőbb felhasználási területe a dinamikus weboldalak készítése. Emiatt a PHP-t jórészt szerver-oldalon használják, bár van parancssori interfésze is, illetve önálló, grafikus felületű alkalmazásokat is létre lehet vele hozni. A nyelvet eredetileg Rasmus Lerdorf alkotta meg 1994-ben, de a ma létező egyetlen (és hivatalos specifikáció híján de facto szabvánnyá vált) PHP implementációt már a PHP Group tartja karban és fejleszti. A PHP a saját licensze alatt kerül kiadásra, a Free Software Foundation így szabad szoftverként tartja számon.

A PHP a legtöbb webserverre, operációs rendszerre és platformra ingyenesen telepíthető. Manapság több mint 20 millió weboldal és egymillió szerver futtat PHP-t, bár a nyelvet használó oldalak száma 2005 augusztusától kezdve folyamatosan csökken. A PHP emellett az Apache webservert egyik legnépszerűbb beépülő modulja.

A PHP fejlődése kezdetén csak CGI-programok halmaza volt. Ezeket Lerdorf néhány Perl szkript lecserélésére írta, amelyeket honlapjának karbantartására (például önéletrajzának megjelenítésére és a látogatottság mérésére) használt. Később ezeket a programokat kombinálta a szintén általa írt Form Interpreter (űrlap-értelmező) alkalmazással - így jött létre a PHP/FI, ami már jóval szélesebb funkcionalitással bírt. Az új, C nyelven megírt változat képes volt adatbázisokhoz kapcsolódni és segítségével egyszerű dinamikus weboldalakat is létre lehetett hozni. Lerdorf 1995. június 8-án adta ki a PHP első nyilvános változatát, hogy külső segítséggel gyorsabbá tegye a hibák megtalálását és a kód további fejlesztését. Az új verziót PHP 2-nek nevezte el és már megtalálhatóak voltak benne a mai PHP alapvető tulajdonságai: a Perl-éhez hasonló változók, az űrlapok kezelése és a HTML-kód

beszúrásának lehetősége. A PHP szintaktikája is hasonló volt a Perl-éhez, de annál jóval korlátoltabb, egyszerűbb és kevésbé egységes volt.

1997-ben Zeev Suraski és Andi Gutmans, két izraeli fejlesztő újraírta az értelmezőt, ezzel megteremtve a PHP 3 alapját - ekkor született meg a PHP új neve, a PHP: Hypertext Preprocessor rekurzív rövidítés is. A fejlesztőcsapat 1997. novemberében, két hónapnyi béta-tesztelés után kiadta a PHP/FI 2-t, majd megkezdődhetett a PHP 3 tesztelése és végül megjelenése. 1998 júniusában Suraski és Gutmans ekkor megint a PHP-mag újraírása mellett döntött, amiből 1999-ben megszületett a Zend Engine. A két fejlesztő ekkor alapította meg a Zend Technologies-t is, ami máig aktívan ellenőrzi a PHP fejlesztését.

A Zend Engine 1.0 által hajtott PHP 4 2000. május 4-én jelent meg. Ezt követte 2004. július 13-án a következő nagy mérföldkőnek számító, az új Zend Engine II-n alapuló PHP 5. Az ötös verzió sok újítást tartalmazott: fejlettebb objektum-orientált programozási lehetőségeket, a PDO (PHP Data Objects) adatbázis-absztrakciós kiterjesztést, és sok teljesítményt növelő javítást is.

A PHP 4 fejlesztése 2007 végén maradt abba, az utolsó megjelent verziója a 4.4.8 verziószámot viselte - a PHP Group ezekhez és a PHP 4 előző változataihoz 2008 augusztusáig biztosít biztonsági frissítéseket. Jelenleg a PHP 5 a nyelv egyetlen aktívan fejlesztett verziója. A hatos verzió bár már fejlesztés alatt áll, még nem jelent meg. 2008. február 4-től a GoPHP5 kezdeményezés keretében több vezető PHP-alapú szabad-szoftver projekt döntött a PHP 4-támogatás megszüntetése mellett, hogy ezzel mozdítsák előre az ötös verzióra való áttérés folyamatát.

A PHP oldalak elkészítésénél a HTML-t gyakorlatilag csak mint formázást használják, ugyanis ezen lapok teljes funkcionalitása a PHP-re épül. Az különbözteti meg a PHP-t a kliens oldali nyelvektől - pl. JavaScript -, hogy a kód a kiszolgálón fut. Nem lehet megállapítani, hogy milyen kód állíthatta elő a kódot. Ezen felül a webszervert be lehet állítani úgy, hogy a PHP feldolgozzon minden HTML fájlt PHP blokkokat keresve, ezek után már tényleg nincs rá mód, hogy kitalálják, mit is rejt egy-egy programod. Ehhez egy ún. interpretert (értelmezőt) használ, amely általában egy külső modulja a webszervernek. A PHP nyelv lényegében nagymértékű kiegészítése a HTML-nek, ugyanis rengeteg olyan feladat végezhető el vele, amelyre az ügyféloldali szkriptek nem képesek (vagy ha igen, korlátozottan). Ilyen például a bejelentkezés, az adatbáziskezelés, filekezelés, kódolás, adategyeztetés, kapcsolatok létrehozása, e-mail küldése, adatfeldolgozás, dinamikus

listakészítés stb. Minden olyan esetben, ahol nagyszámú ismétlődő feladatsort kell végrehajtani (például képek listázása és linkelése, listakészítés stb.), ott ez a programnyelv nagyszerű segítség.

A PHP programok futhatnak közönséges (parancssori) programként is, nem HTML oldalba építve. Ezt azonban ritkán használják. Ezen módot sokszor weblapokkal kapcsolatos időzített folyamatok megvalósításához alkalmazzák, mivel azonos programnyelven, azonos megközelítési módon hajthatók azok végre.

Lehetőség van komplex konzolos alkalmazások írására is, mivel lehetőség van ncurses használatára is.

Windows platformra is írhatók grafikus programok PHP-ben. Ehhez több mód is rendelkezésre áll: PHP-GTK - a sok program által használt Gimp Tool Kit függvényeit és felületi eszközeit lehet elérni.

Winbinder - egy fejlett modul és futtató környezet páros. A standard felületi elemeket teszi elérhetővé, valamint lehetőséget ad egyedi elemek létrehozására. A csomag tartalmaz egy integrált fejlesztői eszközt is, melynek segítségével Drag and Drop módszerrel lehet összerakni a kívánt felületet.

Lehetőség van a php-ben írt források önállóan futtatható programmá alakítására vagyis fordításra. Erre is több módszer létezik, de ezek jelenleg mind erősen a fejlesztési szakaszban vannak.

Van hivatalos megoldás is az utóbbira, ennek lényege, az egyes futásokkor megvizsgálja a forrást és ha az megváltozott generál egy köztes állapotot, mely a forrás fölösleges részeit (kommentek, fölösleges blokk-kezdések, whitespace-ek) kiszűri és generált egy állományt az így kapott változattal. Ha a kód nem változott, betölti a "könnyített" változatot és azt futtatja. Ez jelentheti azt is, hogy az ember által olvasható kódot tokenizálja . Így a feldolgozás egy részét "megspórolva" fut a program. Bizonyos esetekben ez a módszer nem működik, mivel a kód tartalmazhat nem feldolgozható elemeket.

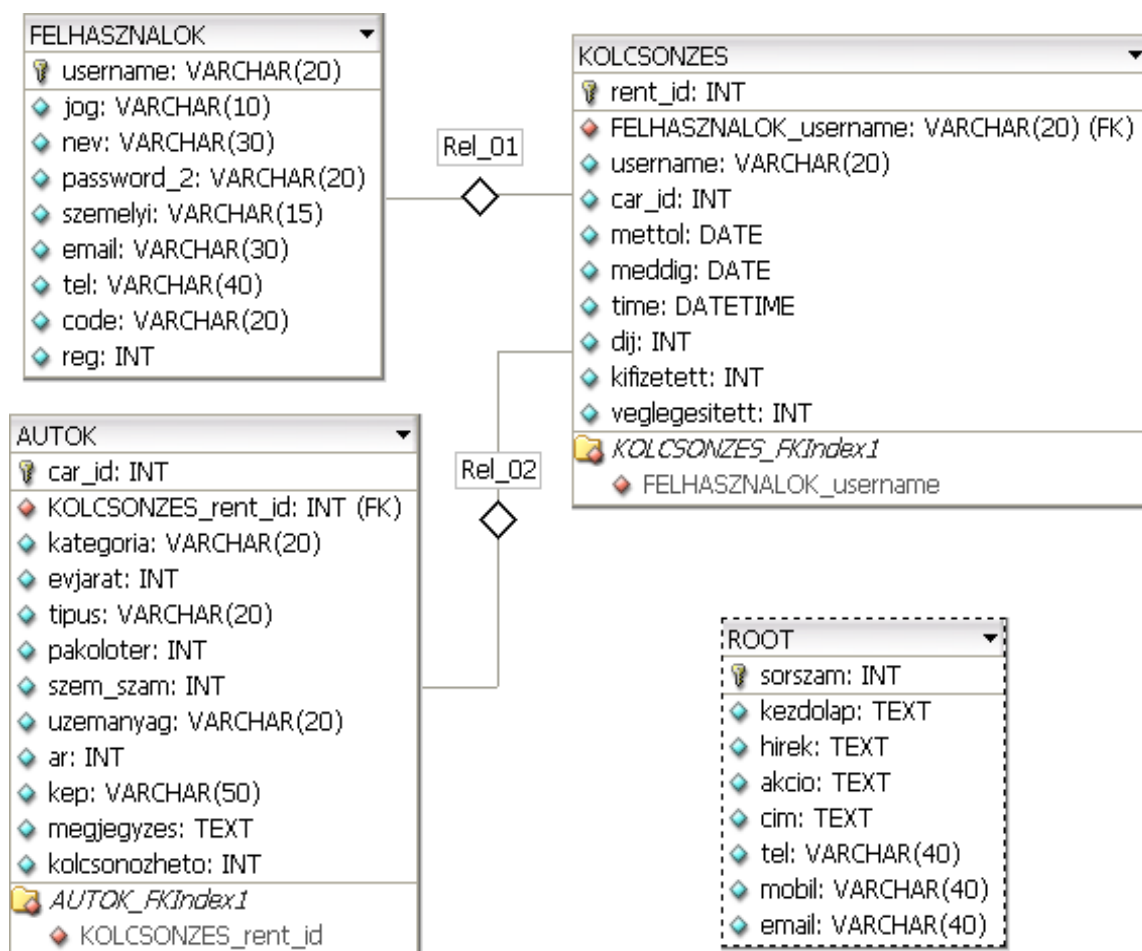
Szintaktikailag könnyű programnyelvről van szó, ugyanis amellet, hogy nem kell az utasítások típusával törődni (néhány speciális eset kivételével), azonkívül az egész nyelv egyszerű. (Ez időnként nehezen megtalálható hibához vezet.)

## II. Az oldal felépítése

Mindenekelőtt meg kell határoznunk, hogy a weboldal mire legyen képes, milyen funkciói legyenek.

Az oldalnak a következő funkciókkal biztosan rendelkeznie kell: kezdőlap, elérhetőség, gépjárművek és azok adatainak listázása, és a karbantartó által feltöltött hírek megjelenítése. Mivel ez egy autókölcsönző weboldala lesz, ezért egyből felmerül a kérdés, hogy lehessen-e online kölcsönözni a gépjárműveket. Mivel szeretnénk ezt a funkciót is biztosítani, ezért feltétlen szükség lesz a látogató és a root felhasználók mellé egy 3. típusú felhasználót is felvenni, aki már regisztrálva van, és akinek jogában áll a kölcsönzés. Ezt végiggondolva tehát feltétlen szükség lesz egy bejelentkező és regisztrációs felületre is.

A root speciális funkciói a következők: a kezdőlap, hírek, elérhetőség módosítása, új gépjármű felvitele, meglévő gépjárművek törlése, kölcsönzések jóváhagyása, illetve törlése, illetve felhasználók törlése(egyéni kérésre). Ezek alapján az elkészített adatbázis a következőképpen néz ki :



Mint látható 4 táblázatomban van, FELHASZNÁLOK, AUTOK, KÖLCSONZES, ROOT. A FELHASZNÁLOK táblában a username az elsődleges kulcs, de meg kell adni a nevet, a jelszót, a személyi- vagy jogosítvány-számot, az email címet a telefonszámot is, ezeken felül a program automatikusan hozzárendeli a jogot (user), a regisztráláshoz szükséges kódot és a reg nevű mezőt, mely azt jelzi, hogy véglegesítve van-e a felhasználó regisztrálása. Az AUTOK nevű táblában tárolom az adatbázisban szereplő gépjárművek adatait. Itt az elsődleges kulcs a car\_id, de kötelező megadni a kategóriát, évjáratot, típust, az üzemanyag fajtáját és a kölcsönzési napi árat. Ezen túl még megtalálható a pakolótér, személyek száma mezők, megjegyzés mezők, melyeket nem kötelező kitölteni. A megjegyzés mező azért TEXT típusú, hogy a plusz információk is bőven elérhessenek. Van még egy plusz mező ebben a táblában, ez pedig a kölcsönözhető mező, mely ha a gépjármű meghibásodott vagy valami más okból kifolyólag átmenetileg nem kölcsönözhető, azt ennek a mezőnek adott 0 értékkel jelezhetjük.

Az utolsó táblában, a ROOT-ban az elsődleges kulcs a sorszám, kívül az egyes menüknek a tartalmát tárolom, ilyenek a kezdőlap, hírek, akció az elérhetőség menük (cím, tel, mobil, email).

A következő oldalon lévő UML az osztályok közötti viszonyokat szemlélteti. Mint látható összesen 17 osztállyal dolgozok, melyből egy abstract osztály (Rent), és 9 valamely másik osztály leszármazottja.

Az egyes osztályok funkciói:

Login: bejelentkezés

Registration: regisztráció

Profil: profil megjelenítése

ProfilMod: profil módosítása

PassMod: jelszó módosítása

Gepjarmuvek: a gépjárművek listázása

rootGepjarmuvek: a root speciális gépjárművekkel kapcsolatos funkciói

Kolcsonzes: a kölcsönzési folyamat

Rent (abstract): a ListaActual, ListaAll és ListaNoSubmit osztályokhoz egyaránt szükséges metódus, melyek a kölcsönzések listázásának különböző típusainak feleltethetőek meg

Felhasznalok: a felhasználók listázása és törlése

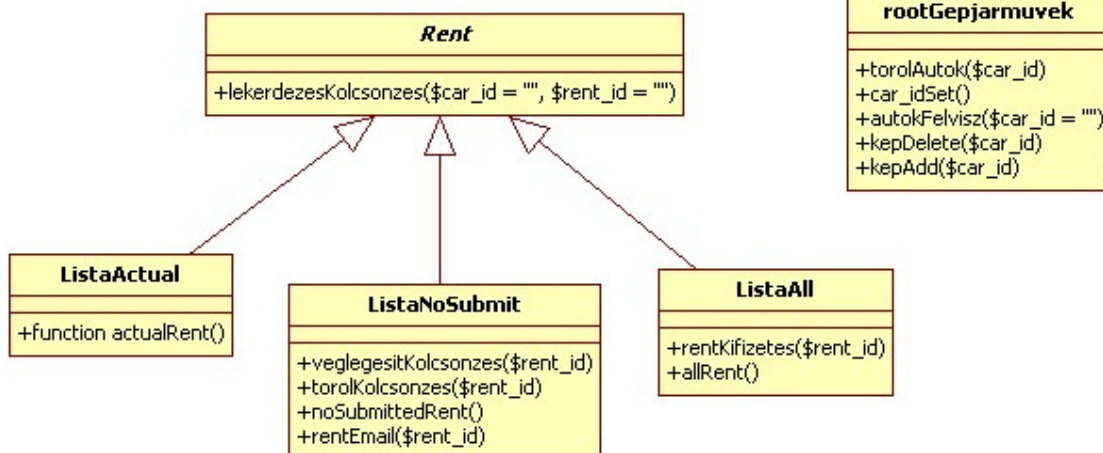
RootLekerdezes: a Root táblához csatlakozik és ennek a leszármazott osztályai az Elerhetoseg, Menuk és a Root.

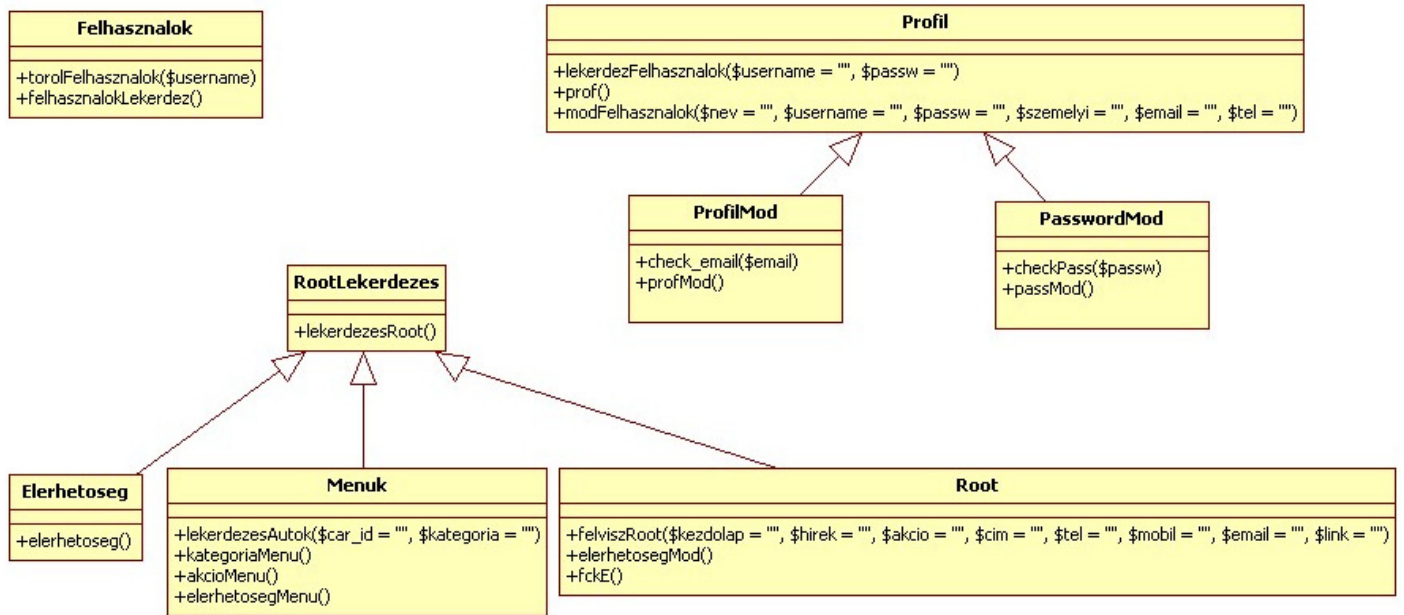
Login
+bejelentkezés() +lekerdezesFelhasznalok(\$username = "", \$passw = "") +log()

Registration
+reg1(\$nev = "", \$username, \$passw, \$szemelyi, \$email, \$tel = "", \$code) +reg2(\$username) +reg3(\$username) +regisztracio() +checkPass(\$passw) +kodgener() +email(\$cimzett, \$username, \$passw, \$kod) +check_email(\$email) +regisztracio2()

Kolcsonzes
+lekerdezesAutok(\$car_id = "", \$kategoria = "") +lekerdezesKolcsonzes(\$car_id = "") +felviszKolcsonzes(\$username, \$car_id, \$mettol, \$meddig, \$dij) +kolcs(\$car_id) +autoFoglalt(\$car_id) +dokLetoltes() +pushfile(&\$file) +kolcsonzesFelvisz(\$car_id) +kolcsonzesVeglegesit() +kolcsonzesLekerdez(\$car_id)

Gepjarmuvek
+lekerdezesAutok(\$car_id = "", \$kategoria = "") +lista() +szelektK() +szelektT() +tablazat()





### **III. Megvalósítás**

#### **Program összes funkciójának betöltése**

1. A fuggvenyek.php fájl-ban az adatbázis elérésére szolgáló metódusban, a 3. sorban kezdődő csatlakozas()-ban értelemszerűen kell kitölteni az eléréshez szükséges adatokat.

2. Ahhoz, hogy az fckeditor képnézegető illetve képfeltöltő funkciója is tökéletesen működjön, a

/scripts/fckeditor/editor/filemanager/upload/php/config.php és a

/scripts/fckeditor/editor/filemanager/browser/default/connectors/php/config.php fájlokban a \$Config['UserFilesAbsolutePath'] változónak meg kell adni a valódi teljes elérési utat.

3. Az osztályok.php -ben a

225. 229. 244. 248. 1747.

sorokban a http://localhost/~marincsak/Szakdolgozat3 hivatkozásrész helyére az aktuális elérési utat kell megadni.

4. Ezzel a felhasználónév és jelszó párossal használhatja a postafiókját, és módosíthatja az Elérhetőség menü alatt lévő Gmail Maps térképet:

felhasználónév: geriszakdolgozat@gmail.com

jelszó : buta13dien

#### **Csatlakozás adatbázishoz:**

A legelső lépés, hogy csatlakozzunk az adatbázishoz. Ezt a resource mysql\_connect ([ string \$server [, string \$username [, string \$password [, bool \$new\_link [, int \$client\_flags ]]]) függvénnyel tehetjük meg, mely megnyit vagy újrahasonít egy MySQL szerver kapcsolatot.. A server a MySQL szerver. Tartalmazhat egy port számot is, pl. "hostname:port" vagy egy helyi socket útvonalát, pl. ":/path/to/socket". Ha a mysql default host PHP direktíva nem definiált (ez az alapértelmezés), akkor az alapértelmezett értéke 'localhost:3306'. SQL safe mode-ban, ezt a paramétert kihagyja, és mindig 'localhost:3306' értéket használja a PHP. A username a

felhasználónév. Alapértelmezett értéke a mysql default user. SQL safe mode-ban, ezt a paramétert kihagyja, és a szerverfolyamat a tulajdonosának a nevét használja a PHP. A password a jelszó. Alapértelmezett értéke amysql default password. Az SQL safe mode-ban, ezt a paramétert kihagyja, és egy üres jelszót használ a PHP. Ha a mysql\_connect() függvényt kétszer ugyanazokkal a paraméterekkel hívjuk meg, akkor nem jön létre újabb kapcsolat; a függvény a már meglévő kapcsolat azonosítóját fogja visszaadni. A new\_link paraméter módosítja ezt a viselkedésmódot, és a mysql\_connect() függvényt új link megnyitására kényszeríti, még akkor is, ha előzőleg a mysql\_connect() függvényt ugyanazokkal a paraméterekkel hívtad meg. SQL safe mode-ban ezt a paramétert kihagyja a PHP. A client\_flags az alábbi konstansok kombinációja lehet: MYSQL\_CLIENT\_SSL, MYSQL\_CLIENT\_COMPRESS, MYSQL\_CLIENT\_IGNORE\_SPACE vagy MYSQL\_CLIENT\_INTERACTIVE. SQL safe mode-ban ezt a paramétert kihagyja a PHP. Sikeres végrehajtás esetén visszaad egy MySQL kapcsolat azonosítót, hiba esetén pedig FALSE-ot. Ha sikeresen csatlakozott a MySQLhez, akkor a mysql\_select\_db(string \$database\_name); függvénnyel tehetjük aktívvá a \$database\_name nevű adatbázist. Innentől kezdve minden mysql kérés ezen az adatbázison fut majd le. Visszatérési értéke true vagy false. Ezek után már csak a karakterkódolást kell beállítanunk (most UTF-8-at használunk), amit a következőképpen tehetünk meg: `mysql_query("SET CHARACTER SET 'utf8'");`

### **SqlSelect és SqlMod**

Azért, hogy ne kelljen minden egyes adatbázis lekérdezés, illetve adatbázis módosító utasítás esetén ne kelljen az adatbázis csatlakozási illetve kérés függvényeket meghívni , létrehoztam ezt a 2 segéd metódust, melyek paraméterül kapnak egy stringet, ami egy sql utasítás, melyet a mysql\_query(\$string) függvénnyel továbbítunk az adatbázisnak. Ha ez az sql utasítás lekérdező utasítás volt , akkor az eredményt beleteszem egy tömbbe, a mysql\_fetch\_array metódu segítségével., melynek a paramétere a kérés eredménye. Majd az SqlMod esetén a visszatérési érték true vagy false a módosítás sikerességének a függvényében, míg az SqlSelect esetén a tömböt adjuk vissza.

<pre>function sqlSelect(\$select){     csatlakozas();     \$query = mysql_query(\$select);     while(\$eredmeny = mysql_fetch_array(\$query)){         \$tomb[]=\$eredmeny;     } return \$tomb;} </pre>	<pre>function sqlMod(\$script){     csatlakozas();     \$query=mysql_query(\$script);     if(!\$query){         return false;     }     else{         return true;}} </pre>
--	---

### **beSession()**

A beSession egy olyan metódus , mely visszatérési értéke 0, ha nem vagy bejelentkezve, 1 ha userként vagy bejelentkezve 2 meg ha rootként. De mi alapján tudja ez a függvény eldönteni?

Többféle lehetőség van arra, hogy a felhasználó bejelentkezés és kijelentkezés közötti viszony (session) alatt az ő adatai legyenek elérhetőek. Ilyen lehetőség pl. egy űrlap rejtett mezőinek használata, vagy az információk URL-ben való tárolása. Ennek viszont az a hátránya, hogy az oldal HTML-kódjából ill. a böngésző location mezőjéből ezek az információk kiolvashatók. Ez ellen az információk kódolásával ,pl : md5-ös kódolással.

Viszont egy nagyobb projekt , hivatkozás esetén ez a módszer sem jó mert elég nehézkesed teszi az adatok kezelését, sőt itt az a probléma is felmerül, hogy az URL-hez fűzhető karakterláncok hossza a böngészők által korlátozottak.

Egy másik módszer a cookie-k használata, mely egy rövid szöveges információ, a merevlemezen tárolódik. Ezekhez az információkhoz az adott oldal bármikor hozzáférhet, akár módosíthatja is. Viszont a cookie-k használatát nem minden böngésző támogatja, illetve sokan nem tartják előnyösnek ezt a módszert biztonsági okok miatt.

Ezek a módszerek amiket említettem eddig kliens oldali megoldások voltak, melynek hátránya az, hogy az információk a felhasználó böngészőjében tárolódnak el, lehetőséget adva az adatok esetleges rossz szándékú manipulálására.

De amit én használtam, az a session-adatok nyilvántartásának egy másik módja, ezek szerver oldali eltárolása. A PHP beépített függvényei segítségével kényelmesen meg lehet oldani a session-kezelést. A munkamenet egy egyedi azonosítót (session id-t) biztosít mellyel az egyik

oldalról a másikra vihetünk információkat. Amikor a felhasználó betölt egy munkamenetet támogató oldalt, kap egy azonosítót, vagy a régebbi látogatásakor kapott azonosító kerül felhasználásra. A munkamenethez kapcsolódó értékek, változók egy ideiglenes fájlban tárolódnak. Az azonosítót általában egy cookie-ban továbbítja, de meg lehet oldani GET típusú lekérdezéssel is (a fent említett biztonsági okok miatt). A `session_start()` függvénnyel lehet elindítani egy munkamenetet, mely generál nekünk egy azonosítót, amit egy cookie-ban elküld. Ha a session elindult, a `session_id()` függvénnyel kérdezhetjük le és módosíthatjuk az azonosítót. Ha egy szöveget adok paraméterül a `session_id()` függvénynek onnantól kezdve az adott szöveg lesz a session id. Ha még nem kaptunk vissza azonosítót, akkor létrejön egy SID nevű állandó Ez akkor hasznos nekünk, ha a felhasználó letiltotta a cookie-kat, mert ha a SID értékét hozzáadjuk az URL-hez, a munkamenetazonosító automatikusan átvételre kerül amikor meghívjuk a `session_start()`-ot. A `session_start()` meghívása után használhatjuk a `$_SESSION` szuperglobális tömböt. Ezt a tárolni kívánt változó nevével indexeljük. Természetesen tömb típusú változókat is tárolhatunk benne. A `beSession()` függvényem a `$_SESSION[„bejelentkezve”]` és a `$_SESSION[„root”]` szuperglobális tömbök értéke alapján adja vissza a fent említett értékek valamelyikét.

<pre>function beSession(){     session_start();     if(\$_SESSION["bejelentkezve"]==true) {         if(\$_SESSION["root"]==true){             return 2;         }         else{             return 1;         }     }     return 0; }</pre>	<pre>function oldalbetolt(\$mit = "") {     if(!\$mit) {         \$alapertelmezett = "kezdolap";     }     else {         \$alapertelmezett = \$mit;     }      if(file_exists(\$alapertelmezett.".php"))     {         include(\$alapertelmezett.".php");     } }</pre>
---	--

## Oldal betöltése

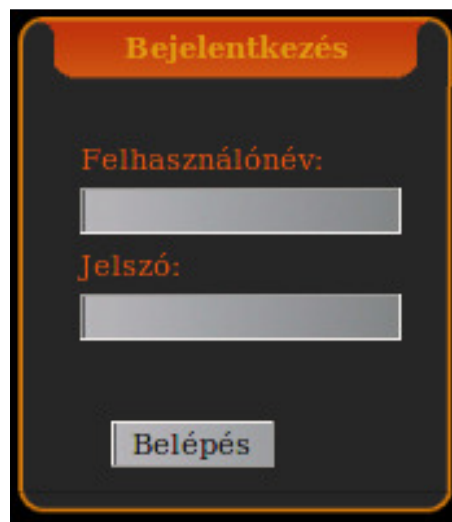
Természetesen az oldal betöltése esetén először az index.php fut le. Ebben a fájlban található az oldal kinézetét, felépítését meghatározó html tagek , melyet később a Kinézet címszó alatt részletezek, itt csak az oldal felépítésének könnyebb megértését szolgálva írok le pár információt. Az oldal kinézete alapvetően 6 részből áll, ahol az első a header, a második a navlist-container a harmadik az a content, a negyedik a post-container, az ötödik a content-belepes , és végül a footer, ami a lap alján található . Most csak a post-containerrel foglalkozunk. Az index.php-nek ezen részében csak ez az utasítás található:

```
print(oldalbetolt($_GET["oldal"])); .
```

Ez az utasítás kiírja az oldalbetolt(string \$változo) visszatérési értékét. De mit is csinál ez az függvény? Megkap egy get paramétert, ha az üres sztring, akkor betölti a kezdolap.php-t, ellenben betölti a megkapott paraméter –esetünkben \$változo- értékének és a “.php” sztringek konkatenációjával létrejött fájlnevet kiterjesztéssel.

## Bejelentkezés

Bejelentkezéskor létrehoztam egy ”felhasznalonev” és egy ”bejelentkezve” nevű elemét ennek a szuperglobális tömbnek és ha a felhasználó root, akkor egy ”root” nevű elemet is , melynek ebben az esetben az értéke true. Ezek alapján tudom meghatározni, hogy be van e jelentkezve az illető vagy sem, és ha igen akkor user vagy root –e. Mint ahogy az előbb említettem, az index.php egyből meghívja a kezdolap.php-t , így azzal találkozunk először. Mivel még SESSION-ben nem tároltunk el semmit sem, ezért nem is lehetünk bejelentkezve, ezért a bejelentkezes() függvény meghívja a log() függvényt, mely lefutása után 2 input mezővel találkozunk (felhasználónév, jelszó). Ezek helyes megadása után tároljuk az említett információkat a SESSION-ben.



The image shows a login form with a dark background and orange accents. At the top, there is a header with the text "Bejelentkezés" in orange. Below the header, there are two input fields. The first is labeled "Felhasználónév:" in orange, and the second is labeled "Jelszó:" in orange. Below the input fields, there is a button labeled "Belépés" in white text on a dark background.

## Kijelentkezés

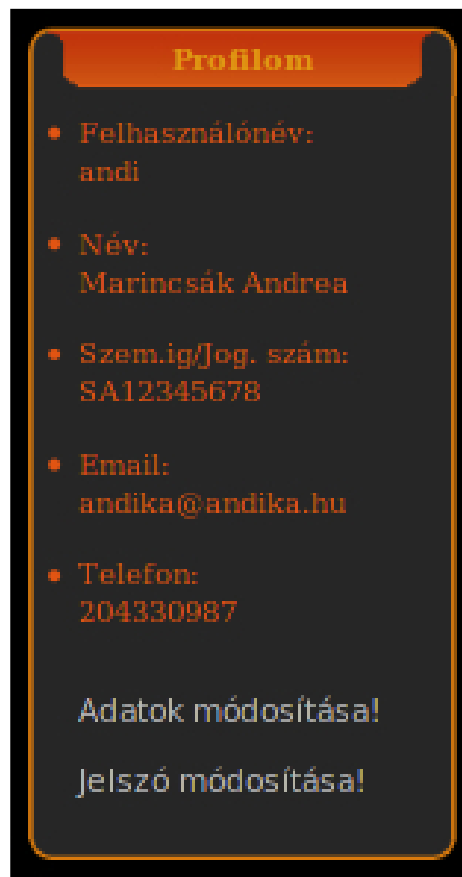
Munkameneteket a `session_destroy()`; függvénnyel törölhetünk, így kijelentkezéskor egyszerűen meghívom ezt a beépített függvényt így törölöm a sessiont, és mivel minden oldal betöltésénél meghívom a `beSession()` függvényemet –leellenőrizve annak jogos használtát– ezért ha ekkor 0 a visszatérési érték, akkor átléptetem a sima látogatók által megtekinthető kezdőlapra.

A `void session_unset ( void )` üggyvény minden éppen bejegyzett munkamenet változót felszabadít.

```
function kijelentkezés(){
    session_start();
    session_destroy();
    session_unset();
}
```

## Profil

Sikeres bejelentkezés esetén lefutnak a Profil osztály metódusai , melyeknek köszönhetően a képernyő jobb oldalán egy profil menü jelenik meg a bejelentkezés menü helyén. Ebben a megadott adatokon kívül található egy 'Jelszó módosítása' és egy 'Adatok módosítása' linket, illetve root esetén egy linket a `www.gmail.com` címre, ahol a beérkező leveleit tudja elolvasni. A 'Jelszó módosítása' link megnyomásának következtében meghívódik a Profil osztály egyik gyermek osztálya , a PassMod osztály, melynek metódusainak köszönhetően 3 input mező jelenik meg előttünk. Az első mezőbe a régi , a másik 2 mezőbe az új jelszó megadása, és a 'Mentés'



**Profilom**

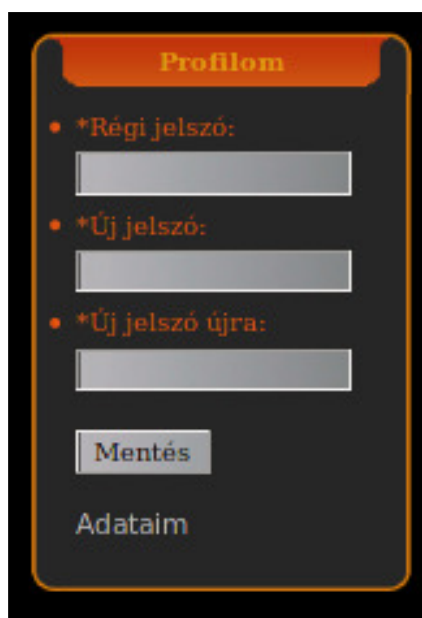
- Felhasználónév: andi
- Név: Marincsák Andrea
- Szem.ig/Jog. szám: SA12345678
- Email: andika@andika.hu
- Telefon: 204330987

Adatok módosítása!  
Jelszó módosítása!

submit gomb megnyomása esetén a checkPass() metódus meghívásával leellenőrzöm annak helyességét. A checkPass(\$passw) metódus visszatérési értéke egy szám ,amit a megadott jelszó karaktereinek gyakorisága alapján generált a függvény. Ha ez a szám nem nagyobb mint 16 akkor a program nem fogadja el a jelszó módosítást, és kiírja a következő üzenetet: "Túl gyenge jelszó!". Ellenben ha az adott jelszó 4 karakternél nem kevesebb , akkor elfogadja és a modFelhasznalok

(\$nev="", \$username="", \$passw="", \$szemelyi="", \$email="", \$tel="") metódus meghívásával a FELHASZNALOK adatbázisban is megváltoztatja a jelszó értékét.

Az 'Adatok módosítása' link megnyomására pedig a ProfilMod osztály hívódik meg. Ekkor a felhasználónéven és a jelszón kívül minden adatot módosítani lehet, viszont csak a jelszó megadásával lehet hitelesíteni. A megadott email címet leellenőrzöm a check\_email(\$email) metódussal, hogy szintaktikailag helyes-e, azaz, hogy 3 részből áll-e ahol a középső egy @ karakter. Ha helyes a jelszó illetve szintaktikailag helyes a megadott email cím, akkor e modFelhasznalok(\$nev="", \$username="", \$passw="", \$szemelyi="", \$email="", \$tel="") metódust hívom meg, mely módosítja a FELHASZNALOK táblát az adatbázisban. Ha ez is sikeres akkor kiírja, hogy „Sikeres módosítás!”.

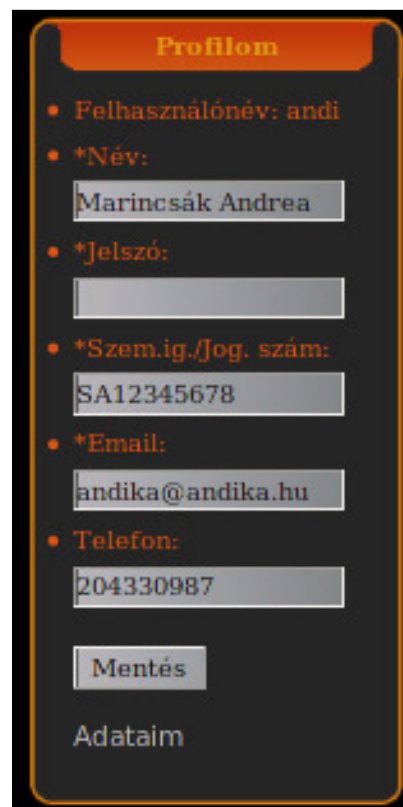


**Profilom**

- \*Régi jelszó:
- \*Új jelszó:
- \*Új jelszó újra:

[Adataim](#)

Jelszó módosítása



**Profilom**

- Felhasználónév: andi
- \*Név:
- \*Jelszó:
- \*Szem.ig./Jog. szám:
- \*Email:
- \*Telefon:

[Adataim](#)

Profil módosítása

## Regisztráció

Regisztrálni csak nem bejelentkezett állapotban és csak user lehet, rootot nem. A 'Regisztráció' menüre kattintva meghívódik a Registration osztály. Először a regisztracio() metódus fut le, melynek hatására egy űrlap jelenik meg a képernyőn. Az űrlap adatai közül a telefonszám megadásán kívül az összes többi adat megadása kötelező. Miután elküldte az adatokat először leellenőrzöm, hogy a megadott felhasználónév rövidebb-e mint 4 karakter illetve, hogy foglalt-e már és ha igen akkor ezt kiírom hibaiüzenetbe. Majd leellenőrzöm, hogy a 'Jelszó' és a 'Jelszó újra' mezők megegyeznek e. És végül leellenőrzöm, hogy a megadott jelszó hossza kisebb-e mint 4 karakter, illetve, hogy az erre a jelszóra meghívott checkPass() függvény visszatérési értéke kisebb-e mint 16, ha igen akkor szintén hibaiüzenetet írok ki. Ellenben ha ezen is átmennek a megadott adatok akkor meghívom a kodgener() függvényt mely generál egy 15 karakterből álló sztringet a karaktereket randomizálva, mely sztring egyben a visszatérési értéke is ennek a függvénynek. Ezek után meghívom az email(\$\_POST["email"],\$\_POST["username"],\$\_POST["passw"],\$kod) függvényt, ahol a \$\_POST["email"] az imént a regisztráló által megadott email cím, a \$\_POST["username"] az imént kiválasztott felhasználónév a \$\_POST["passw"] a jelszó és a \$kod a kodgener() visszatérési értéke. Ez a függvény a check\_email(\$email) függvény segítségével megállapítja, hogy szintaktikailag helyes-e a megadott email cím, és ha nem, akkor false értékkel tér vissza és a regisztráció sikertelenül ér véget. Ha true értékkel tér vissza a check\_email(\$email) akkor szintaktikailag helyes a megadott email cím, és az email metódus elküld erre a címre egy regisztrációs levelet. Az email küldés egy beépített függvény segítségével történik, mely függvény a bool mail ( string \$to , string \$subject , string \$message [, string \$additional\_headers [, string \$additional\_parameters ]]). A string \$to érték a címzett email címét jelenti, a string \$subject a levél tárgyát, a string \$message a levél szövegét. Lehet még további paramétereket megadni, mint például a string \$additional\_headers, ami az email fejlécét állítja be a megadott értékre.

## Kölcsönözzön autót

a RENTAL AUTÓKÖLCSÖNZÓNÉL!



Kezdőlap

Regisztráció

Hírek

Gépjárművek

Elérhetőség

### Akciók

Június!!!

A kikölcsönzött 7. nap után

5 % kedvezmény!

### Kategóriák

- Mini
- Családi
- Mikrobuszok

### Elérhetőség

- Debrecen Egyetem tér 1.
- 06(52)455-413
- 06(20)436-09-54
- geriszakdolgozat@gmail.hu

### Regisztráció

*Név:	<input type="text"/>
*Felhasználónév	<input type="text"/>
*Jelszó:	<input type="text"/>
*Jelszó újra:	<input type="text"/>
*Jogosítvány:	<input type="text"/>
*Email cím:	<input type="text"/>
Tel:	<input type="text"/>

Adatok küldése!

### Bejelentkezés

Felhasználónév:

Jelszó:

Belépés

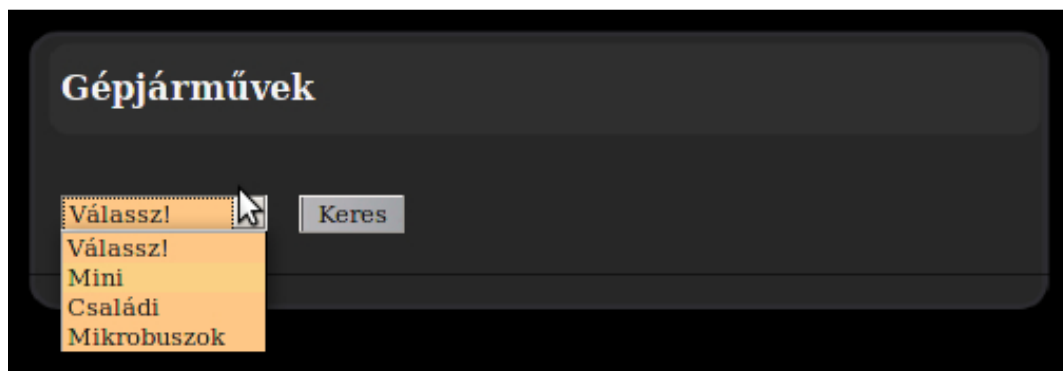
created by gregike gregike@gmail.com

A regisztrációs levélben megtalálható a felhasználó username illetve password adata és az imént generált kód. A levélben található egy link melynek megnyomásával törlődnek az adatbázisból a felhasználó adatai, így ha az email címet nem a tulajdonosa adta meg, akkor a tulajdonosa 1 gombnyomással törölheti magát. Ellenben ha ő kíván regisztrálni akkor szintén egy link segítségével véglegesítheti regisztrációját. Az email sikeres elküldése után az email metódus true értékkel tér vissza az őt meghívó regisztracio() metódushoz. Ezután meghívódik a reg1 metódus a megadott adatokkal, mely felviszi a felhasználó adatait a FELHASZNALOK táblába, és a 'veglegesítve' mező értékét 0-ra állítja. Mindezek sikeres elvégzése után meghívódik a regisztracio2 metódus, mely átléptet egy másik oldalra, ahol az emailben megkapott kód bemásolásával véglegesítheti a regisztrációját a felhasználó. Az

adatbázisban tárolt kód és az itt megadott kód egyezése esetén egy 'Bejelentkezés' link jelenik meg, mely a kezdőlapra navigál.

### **Gépjárművek listázása**

A gépjárművek menü megnyomására a Gepjarmuvek osztály lista() metódusa hívódik meg. Ez először meghívja a szelektK() metódust, melynek következtében egy html select jelenik meg előttünk, ahol kiválaszthatjuk a kívánt kategóriát, majd meghívja a szelektT() metódust, ahol az autó típusát választhatjuk ki egy html select segítségével.



### **Gépjárművek listázása**

Ha kiválasztottuk a típust is akkor a lista() metódus meghívja a tablázat() metódust. Ez a metódus először lekérdezi az AUTOK táblázatból az autókat, illetve megvizsgálja, hogy a felhasználó milyen jogokkal rendelkezik. Erre azért van szükség mivel egyes funkciók innen érhetőek majd el a usereknek illetve a rootnak is. Mindezek után megjelenik előttünk annyi táblázat, ahány a kiválasztott kategóriával és a kiválasztott típussal megegyező gépjármű szerepel az AUTOK táblázatban.

## Gépjárművek

[Gépjárművek listázása](#)

[Új gépjármű hozzáadása](#)

Válassz!

Keres

Válassz!

Keres

Mini

Renault Clio 1.2

*Kölcsönözhető!*

Autó azonosító száma:	2.
Évjárat:	2006
Csomagtér:	190 liter
Személyek száma:	5
Üzemanyag:	dízel
Napi ár (HUF):	5500



[Adatok szerkesztése!](#)

Törlés:

Kijelöltek törlése

### Gépjármű kilistázva rootként bejelentkezve

Ezen táblázatok tartalmazzák az egyes gépjárművek adatait. Ha van kép kapcsolva az adott gépjárműhöz, akkor az a gépjárműhöz tartozó táblázat mellett megjelenik. A egyes gépjárművekhez tartozó adatok egymás alatt helyezkednek el. A rootnak ezen túl minden egyes gépjárműhöz tartozik egy 'Adatok szerkesztée!' link és egy 'Törlés' checkbox. A link megnyomására az URL-ben továbbítom a gépjárművet azonosító car\_id-t , így fogom eldönteni, hogy mely gépjármű adatait kell akarja a root szerkeszteni. A törlésnél használt

checkboxok esetében egy `torles[]` nevezetű tömböt használok. Megvizsgálva ezt a tömböt, ha az *i.* gépjárműnél bepipálja a `root` ezt a checkboxot, akkor annak értéke a `car_id`-je lesz, és akkor meghívom erre a `torolAutok($car_id)` függvényt, mely az AUTOK táblából törli ezen `car_id`-hez tartozó gépjármű összes adatát és meghívja a `car_idSet()` metódust. A `car_idSet()` metódus beállítja a gépjárművekhez tartozó `car_id`-t a törlés után.

### **Gépjárművek kölcsönzése**

A kölcsönzést a Gépjárművek menüből érhetjük el úgy, hogy a fent említett módon kilistázzuk az azonos típusú járműveket, és ha a látogató be van jelentkezve akkor mindegyik kilistázott gépjármű esetén talál egy 'Gépjármű kölcsönzése' linket. Ennek megnyomására meghívódik a `Kolcsonzesek` osztály `kolcs()` metódusa. Ebben a metódusban csak 3 metódushívást találhatunk. Az első a `kolcsonzesLekerdez($car_id)`, mely létrehozza ugyanazt a táblázatot amit azon az oldalon talált a felhasználó, amelyikről idekerült. Majd a `kolcs()` metódus meghívja először az `autoFoglalat($car_id)` metódust, mely kiírja egy táblázatban, hogy mely időpontokban foglalt már az autó, majd a `kolcsonzesFelvisz($car_id)` metódust. Ez létrehoz egy linket, amelynek megnyomására egy másik oldal nyílik, ahol megnyitódik a `kolcsonzesi_feltetelek.pdf`. Ennek elfogadására egy checkboxot hoz létre, melynek az időpont elküldésekor ki kell lennie pipálva vagy különben hibát ír. Alapból üres ez a checkbox. Ugyancsak ez a metódus hoz létre 2 input mezőt, ahová a kölcsönzés kezdetének és végének az időpontját kell megadni. Ennek szintaktikailag helyes megadását segíti egy beépített javascript, a jquery. A JQuery egy JavaScript könyvtár (Keretrendszer, Eszköztár), amivel egyszerűsödnek a JavaScript feladatok a HTML dokumentumokon, mivel számtalan megoldást készen kapunk benne. De a legnagyobb előnye, hogy a jól megírt produktum, nagy valószínűséggel az összes böngészőben fog futni.

Jellemzői:

- Diszkrét javascript, ezért kódolásra nevel
- Böngésző 'független' JavaScript fejlesztéseket készíthetünk.
- Rendkívül kisméretű, de nagy tudású keretrendszer (tömörítve 15 KB).
- CSS 1-3 kompatibilis.
- Nem ad kész megoldásokat, a programozó nem hiányozhat a gépezetből.
- Rengetek kész Plugin érhető el hozzá
- Rendkívül jól dokumentált, több könyv is elérhető hozzá.
- Felhasználói interfészt is elérhető hozzá.

- Folyamatos fejlesztés alatt van.
- Több nagy projekt a JQuery mellett tette le a voksát.
- Könnyen tanulható, kevés kóddal tudunk látványos produktumokat készíteni.

Visszatérve, az intervallum megadásának helyességét vizsgálom és ha hibásan van megadva az intervallum, akkor hibaüzenetet írok ki. Ekkor is egy beépített függvényt használok, mégpedig az `int strtotime ( string $time [, int $now ] )`-t. A függvény egy stringet vár, amiben egy angolul írt dátum van, és Unix időbélyeggé (1970 január 1. 00:00:00 GMT óta eltelt másodpercek száma) konvertálja, a `now` paramétert, vagy ha az nincs megadva, az aktuális időpontot figyelembe véve. Ha a függvény nem tudja értelmezni a karakterláncot, -1-gyel tér vissza. Szóval a dátumot átkonvertáltam a mostani időpont és a megadott időpont közötti másodpercek számává. Az így átkonvertált 2 megadott, az intervallumot leíró dátum közötti időkülönbségét másodpercekben kapjuk meg.

Végigvizsgálom az ugyanehhez a gépjárműhöz tartozó véglegesített kölcsönzések időintervallumait, hogy beleesik-e valamelyikbe a megadott intervallum. Ha igen, akkor szintén hibaüzenetet írok ki. Ha a megadott dátumok minden kritériumnak megfelelnek, akkor meghívódik a `kolcsonzesVeglegesit()` metódus. Ez egy táblázatban visszaadja azokat az adatokat amiket az imént jóváhagyott a kölcsönző személy. Ebben a táblázatban már megtalálható a megadott intervallumból kiszámolt napok számából és az autóhoz tartozó napi díj szorzata alapján kiszámolt kölcsönzési költség is. Ha mindent rendben talál akkor a 'Kérés véglegesítése!' gomb megnyomásával véglegesítheti azt. Ekkor meghívódik a `public function felviszKolcsonzes($username,$car_id,$mettol,$meddig,$dij)` metódus, mely felviszi a KOLCSONZESEK táblába a kölcsönzést úgy, hogy beállítja a 'veglegesitve' mezőt 0-ra. Ha ez sikeresen megtörtént, akkor a program kiírja a következő üzenetet: "Sikeresen felvittük az adatbázisunkba a kérést, melynek elfogadásáról emailben értesítjük!", ellenkező esetben hibaüzenetet ír ki.

## Gépjárművek

### Renault Clio 1.2

Kölcsönözhető!

Autó azonosító száma:	2.
Évjárat:	2006
Csomagtér:	190 liter
Személyek száma:	5
Üzemanyag:	dízel
Napi ár (HUF):	5500



Ezekon a napokon már kikölcsönözött:

Mettől	Meddig
2009-03-11	2009-03-20
2009-04-16	2009-04-24
2009-05-04	2009-11-04
2009-04-05	2009-04-10
2009-04-10	2009-04-14

A kölcsönzési feltételek letölthetők [ITT](#).

A kölcsönzési feltételeket elfogadom :

Kérjük adja meg a kölcsönzés idejét:

Kölcsönzés kezdete :

11/12/2009

Kölcsönzés lejárt :

Időpont mentése!

Válassz!

Keres

November 2009						
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

Kölcsönzés 1. lépés

## Gépjárművek

[Vissza](#)

Az Ön kérése a következő :

Kölcsönző személy	Autó azon. száma	Autó típusa	Kölcsönzés kezdete	Kölcsönzés vége	Fizetendő összeg
andi	2	Renault Clio 1.2	11/12/2009	11/27/2009	82500 Ft

Kérés véglegesítése!

Válassz!

Keres

### Kölcsönzés 2.lépés

## Menük

Ebben a részben a képernyő bal oldalán található akció, kategóriák és elérhetőségek menük egyaránt a Menuk osztály egyes metódusait használják. Az akciók kiírását az akcioMenu(), a kategóriák kiírását a categoriaMenu() míg az elérhetőség kiírását az elerhetosegMenu() végzi. Mindegyik metódus használja a RootLekerdezes osztály lekerdezesRoot metódusát, mely a ROOT táblának a sorait adja vissza eredményül.

### Akciók

Június!!!  
Minden kölcsönzésre  
5% kedvezmény!!!

### Kategóriák

- Mini
- Családi
- Mikrobuszok

### Elérhetőség

- Debrecen
- 06(52)412-412
- 06(20)412-41-24
- geriszakdolgozat@gmail.hu

Menü boxok

## Kezdőlap, hírek, elérhetőség

A fenti menüpontok közül az Elérhetőség menüpontot kiválasztva az Elerhetoseg osztály hívódik meg, így lefut a konstruktora. A konstruktor először lekérdezi a ROOT táblából az adatokat a lekerdezesRoot() metódus meghívásával, majd ezt felhasználva kiíratkja a 'cím' , 'tel', 'mobil' , 'email' értékeket ha azok meg vannak adva. Ezen túl még található egy Google Mapset használó térkép is az oldalon, mellyel később foglalkozunk.

Az index.php-ről eleve átugrunk a kezdolap.php-re, amit máskülönben a Kezdőlap link lenyomásával érhetük el. Ekkor ha nem rootként vagyunk bejelentkezve, akkor a lekerdezesRoot függvény hívódik meg, mely visszatérési értékének 'kezdolap' mezőjét írattjuk ki. Hasonló a helyzet a Hírek menü megnyomásánál, itt a hirek.php fut le, ahol szintén a lekerdezesRoot függvény fut le , melynek visszatérési értékének 'hirek' mezőjét írattjuk ki. Root speciális funkciói

**Kölcsönözzön autót**  
a RENTAL AUTÓKÖLCSÖNZŐNÉL!

Kezdőlap   Regisztráció   Hírek   Gépjárművek   Elérhetőség

**Akciónak**  
Június!!!  
Minden kölcsönzésre  
5% kedvezmény!!!

**Kategóriák**

- Mini
- Családi
- Mikrobuszok

**Elérhetőség**

- Debrecen
- 06(52)412-412
- 06(20)412-41-24
- gersizakdolgozat@gmail.hu

**Üdvözljük honlapunkon!**

Alsó -, közép- és nagykategóriás autó, kisbusz, mikrobusz egyaránt elérhető cégünknel autóbérlés céljából, kedvező feltételekkel.

Az autókölcsönzés szolgáltatás nálunk több, mint hogy bérel egy autót. Amint meg az autóval együtt nyújtunk Önnek: casco biztosítás, nemzetközi asszisztansz szolgálat. Minden autónkat elláttuk éves autópályamatricákkal, így Önnek nem kell a magyar autópályák használatáért fizetnie. A kiszállítás és visszavétel Budapesten, illetve a Ferihegyi repülőtéren 08:00 18:00 között kedvező díjon rendelhető meg.

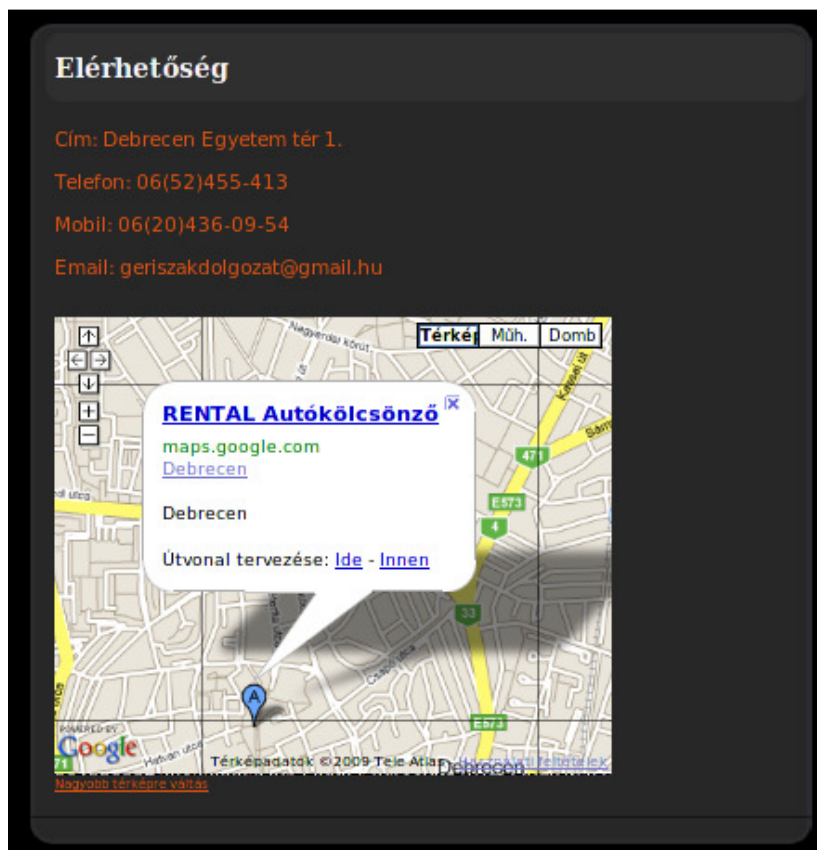
Kitűnő műszaki és esztétikai állapotban lévő autóink, és az autóbérlés szolgáltatásunkhoz kapcsolódó egyéb szolgáltatásaink kielégítik a modern fogyasztói igényeket. A Recent Car autókölcsönző Kft. tagja a Magyar Autókölcsönzők Szövetségének, ami egyben minőségi garanciát is jelent. Felsőkategóriás autóbérlés szolgáltatásunkat ugyanakkor elérhető áron kínáljuk.

**Bejelentkezés**

Felhasználónév:

Jelszó:

created by gregike gregike@gmail.com



Elérhetőség

## **Beállítások**

Ha a felhasználó rootként lép be, akkor is a kezdolap.php fut le először, de az aktív menü neve most nem Kezdőlap, hanem Beállítások. Ebben 4 linket találunk a következő sorrendben:

„Kezdőlap módosítása!

Hírek módosítása!

Elérhetőség módosítása!

Akciók módosítása!”

Mind a kezdőlap, hírek és akciók módosítása link megnyomására az fckE() metódus hívódik meg, melyről a következő alcímben fejtem ki a tudnivalókat. Az elérhetőség módosítása link

megnyomására az `elErhetosegMod()` metódus hívódik meg, mely először meghívja a `ROOT` tábla tartalmát lekérdező `lekerdezesRoot()` függvényt, majd ennek segítségével ha van érték a 'cím', 'tel', 'mobil', 'email' mezőkben akkor azok értékeivel kitölti a hozzájuk tartozó input mezőket. Tehát ezen az oldalon a `root` az elérhetőséget adhatja meg, ami nem csak az Elérhetőség menüben hanem a képernyő bal oldalán található Elérhetőség box értékét is befolyásolja. Az Elérhetőség menüben található még továbbá egy Google Maps alkalmazás, mely a cég földrajzi elhelyezkedését mutatja meg. Ezt nem ezen a honlapon, hanem a Google Maps honlapján a Gmail felhasználónevével és jelszavával változtathatja meg.

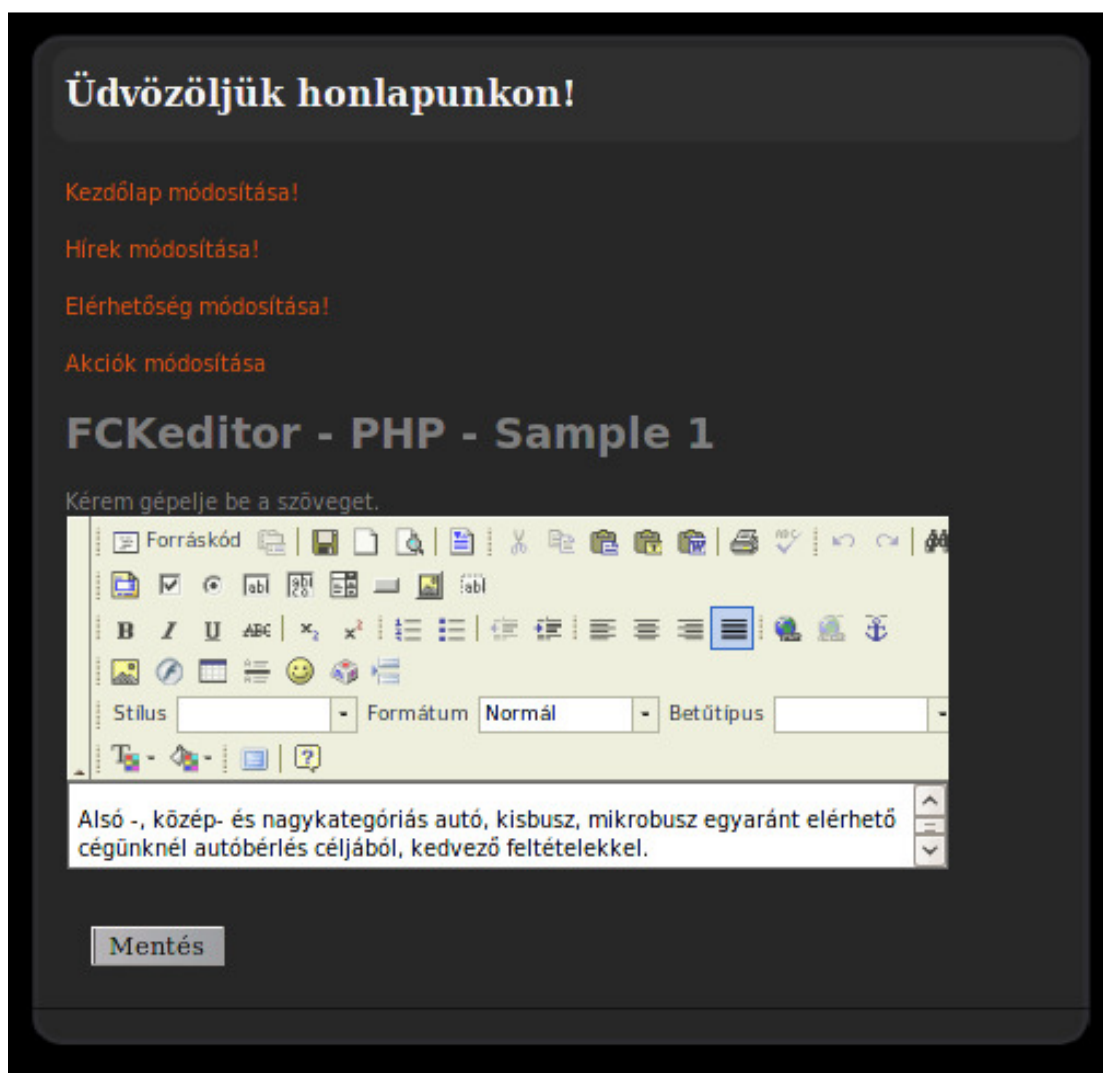
Az `Iframe` (`inline frame`) egy HTML elem amely lehetővé teszi más HTML dokumentumok beágyazását a fő dokumentumba. Az `Iframe` méretét az HTML anyalapban kell meghatározni, és így az anyaoldal már megjelenhet, amíg az `Iframe` még letöltés alatt van. Az `Iframe` egy `inline` képhez hasonlóan viselkedik és a felhasználó képes kigörgetni a képből. Másrészt az `Iframe`-nek lehet saját görgetője is, amely független az anyaoldal görgetőjétől. Míg a sima `frame`-eket általában arra használják, hogy logikailag kisebb egységekre osszák velük a weboldalakat, az `Iframe`-eket inkább arra, hogy segítségükkel más weboldalokról beollózzanak tartalmakat (pl. hirdetések) az aktuális weboldalba.

A beágyazott dokumentumot le lehet cserélni anélkül, hogy az anyaoldalt újra kéne tölteni, a `target` attribútum használatával a HTML horgony (`anchor`) elemében, vagy JavaScript segítségével. Ez lehetővé teszi interaktív webalkalmazás létrehozását, gyakran használják AJAX alternatívájaként vagy vele pl. Flash nélküli fájlküldésre, cross-domain kikerülésére. Legújabban a Mozilla Firefox, az Opera és a Microsoft Internet Explorerböngészőkben bevezették a `contentEditable` és a `designMode`, fogalmakat, melyek lehetővé teszik az `Iframe`-ben beágyazott HTML oldal szerkesztését. Ezt a szolgáltatást használják az olyan `Iframe`-ben elhelyezett rich text (WYSIWYG) szövegszerkesztők, mint például az `FCKEditor` (amiről majd később beszélek).

### **FCKEditor**

Az `FCKEditor` egy nyílt forráskódú WYSIWYG szövegszerkesztő melyet weblapokba szoktak beágyazni. Lehetőséget ad a textarea kiváltására, szinte az összes olyan funkció megtalálható benne, mint egy normális szövegszerkesztőben, pl: Microsoft Wordben. Beállíthatjuk a szöveg színét, méretét, háttérszínét, betűtípusát ... Az alapkódját JavaScriptben

írták. A honlapokba való beágyazása viszont kicsit körülményes, de azért megoldható. Tehát ezzel a beépített szövegszerkesztővel sokkal jobban megoldhatóak a Kezdőlap, Hírek, Akciók formázása mint bármi mással, ezért is választottam ezt a fajta lehetőséget. Az FCKEditorral képet is feltölthetünk a szerverre, mint azt a kezdőlapon is látni lehet.



Kezdőlap módosítása

### Új gépjármű hozzáadása

A Gépjárművek menüben 2 lehetőség közül választhat a root. Az egyik az „Új gépjárművek hozzáadása” a másik a „Gépjárművek listázása”. A root gépjárművekkel kapcsolatos speciális funkcióival a Gepjarmuvek osztály rootGepjarmuvek alosztálya foglalkozik.

Az „Új gépjármű hozzáadása” link megnyomására az autokFelvisz() függvény hívódik meg. Ebben 2 form található. Az egyik az új kategória hozzáadásáért felelős, a másik a többi adat hozzáadásáért. Mindenféleképpen úgy akartam megcsinálni, hogy ne téveszthessen a root, nem akartam lehetőséget adni arra, hogy elgépeljék a kategória nevét, így új kategóriát hozva létre. Ezért aztán a meglévő kategóriákat kiszelektálva választhat belőlük, hogy melyikhez ad hozzá egy újabb járművet. Viszont felvetődött a kérdés, hogy hogyan tud így új kategóriát hozzáadni. Erre egy másik formot kellett létrehoznom, még a többi adat megadása előtt. Az így létrehozott új kategória sehol sem tárolódik, csak a következő formnál, ahol kiválaszthatja a root a kategóriát. A található még továbbá 5 input mező, amelyek közül a típust az évjáratot a napi árat kötelező megadni, és található még 1 szelekt, 1 textarea és egy checkbox is. A szelektből kötelező kiválasztani az üzemanyag típusát a checkbox pedig a kölcsönözhetőségére utal, ha ki van pipálva akkor ki lehet kölcsönözni, ha nem akkor nem (de ha nem lehet kikölcsönözni akkor is listázásra kerül, de ki van írva, hogy nem kölcsönözhető). Erre akkor lehet szükség, ha valamelyik gépjármű meghibásodott ezért nem lehet jelenleg kölcsönözni. A textarea a további információk megadására szolgál, kitölteni nem kötelező. Ha az adatok amelyeket kötelező megadni ki vannak töltve, akkor a mentés következtében meghívódik a felviszAutok függvény, mely felviszi az új gépjármű adatait az AUTOK táblába úgy, hogy a car\_id a kilistázott autók száma+1 -el egyezik meg.

### **Gépjárművek adatainak szerkesztése**

Mint említettem a Gépjárművek menüben lévő 2 link közül az egyik a Gépjárművek listázása. Ennek hatására szinte változtatás nélkül fut le ugyanúgy a Gepjarmuvek osztály lista() metódusa, melynek hatására kiválaszthatunk egy gépjárművet. Az aprócska változtatás csupán annyi, hogy ellenőrzöm, hogy root-e a felhasználó aki belép, és ha igen akkor minden egyes kilistázott autónál 2 új funkció található. Az egyik egy checkbox, melynek kipipálásával törölhetjük a gépjárművet és a hozzá tartozó információkat, képet az adatbázisból illetve a szerverről. A másik egy „Adatok szerkesztése” link, melynek megnyomására egy az új gépjármű hozzáadásánál található táblázattal találkozhatunk, mely ki van töltve az eddig megadott információkkal. Mindent megváltoztathatunk, nincs semmi olyan adat ami rögzített lenne, ugyanis a car\_id egy a felhasználó által nem befolyásolható adat. Például itt tehetjük újra kölcsönözhetővé a gépjárművünket meghibásodás után. Ami

szembetűnhet az a táblázat felett található 2 link , a „Meglévő kép lecserélése!” és a „Meglévő kép törlése!”. A kép törlése linkre kattintva a `kepDelete` függvény hívódik meg, mely először a szerverről, majd az adatbázisból törli a képet az `sqlMod` segítségével. A szerverről a `bool unlink ( string $filename [, resource $context ] )` beépített függvánnyel tehetjük meg, mely `Törli` a `filename` paraméterben megadott fájlt. Hasonló a Unix C `unlink()` függvényéhez. Siker esetén `TRUE` értékkel tér vissza, ellenkező esetben `FALSE` értéket ad.

**Gépjárművek**

[Gépjárművek listázása](#)  
[Új gépjármű hozzáadása](#)

[Gépjármű szerkesztése!](#)  
[Meglévő kép lecserélése!](#)  
[Meglévő kép törlése!](#)

Új kategória hozzáadása:

\*Kategória:  ▾

\*Típus:

\*Évjárat:

Személyek száma:

Csomagtér:  liter

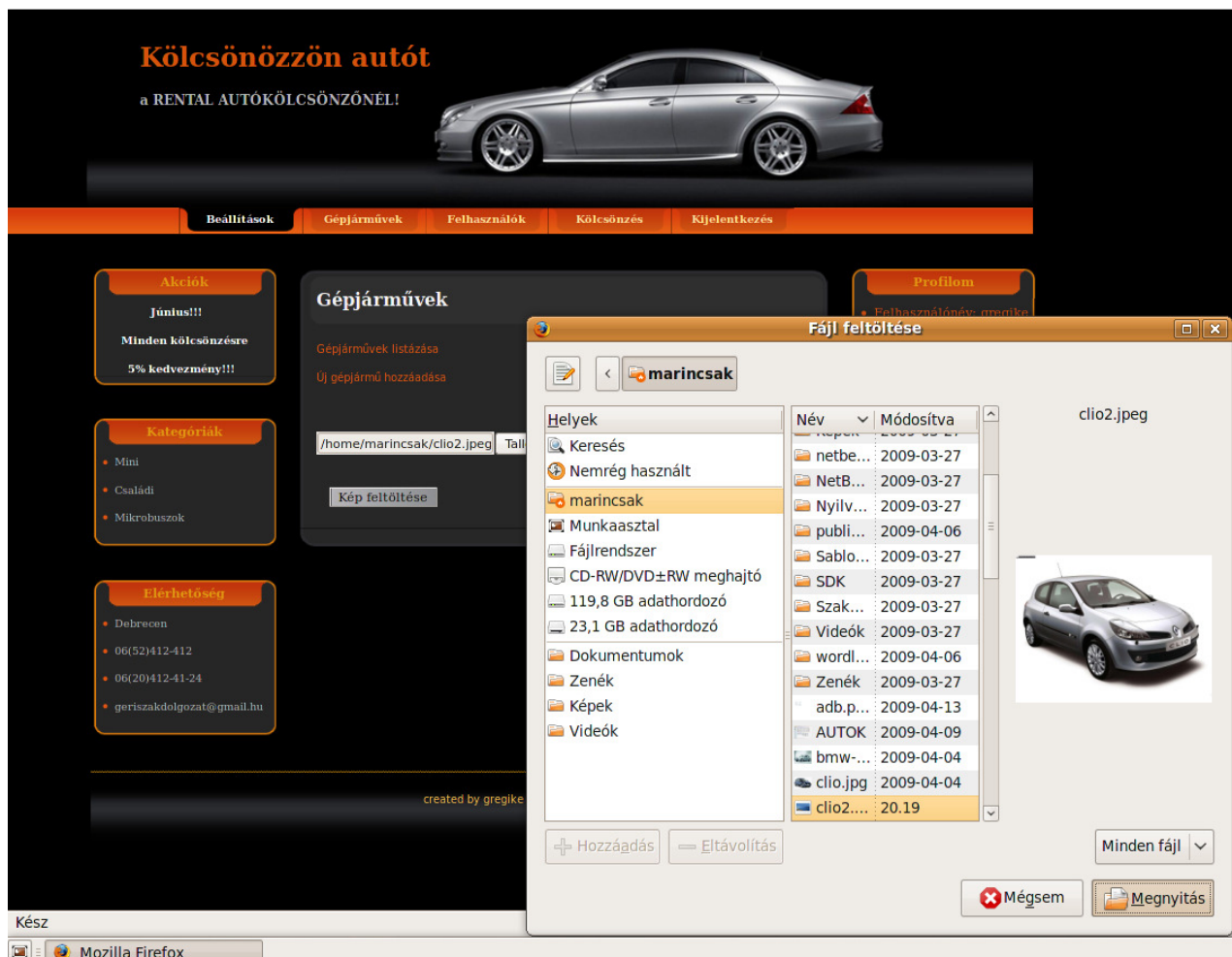
\*Üzemanyag:  ▾

\*Naplár:  HUF

Megjegyzés:

Kölcsönözhető:

Meglévő gépjármű szerkesztése (root)



### Képfeltöltés

A meglévő kép lecserélése link hatására először a `kepDelete` hívódik meg, majd a `kepAdd` metódus és egy új ablakra ugrunk, ahol kitallózhatjuk a feltöltendő képet. Ezzel a 'láthatatlan' input mezővel `<input type="hidden" name="MAX_FILE_SIZE" value="10000000" />` adhatjuk meg a feltöltendő fájlmaximális méretét.

A képfeltöltésnél szükségünk van egy másik php fájlra, esetünkben az `uploader.php`-re. Itt az `strtok` beépített függvény segítségével meghatározom a kiterjesztését a feltöltendő fájlnek, majd a `basename` beépített függvény segítségével létrehozom az új fájlnevet amelynek neve a `car_id`-vel kiterjesztése pedig a feltöltendő kép kiterjesztésével egyezik meg.. Tehát a beépített függvény a következő : `string basename ( string $path [, string $suffix ] )`. A megadott karakterlánc tartalmazza a fájl elérési útvonalát. Ez a függvény ebből visszaadja a fájl nevét. Ha a fájlnev a `suffix` paraméter értékével végződik, ezt a függvény levágja. Windows alatt a

perjel (/) és a fordított perjel (\) is használható könyvtárelválasztó karakterként. Más környezetekben, csak a sima perjel (/) használható.

A `bool move_uploaded_file ( string $filename , string $destination )` beépített függvény segítségével tudjuk ténylegesen a feltöltendő fájlt a szerverre helyezni. Először is ez megállapítja, hogy érvényes-e a megadott fájlnev, majd leellenőrzi a megadott helyet ahová fel kell töltenie, és ha ezek érvényesek, csak akkor végzi el a műveletet. Ha ne érvényesek akkor a visszatérési érték FALSE, ellenben TRUE. Figyelnünk kell arra, hogy a mappát amibe helyezni szeretnénk a megadott fájlt lehessen írni, ellenben a függvény hibüzenetet ír ki és FALSE ad vissza. Ha sikeresen feltöltődött a szerverre a fájl, akkor meghívódik a `modAutok` metódus, mely ezt az adatbázisban is jelzi, az AUTOK táblában.

### Kölcsönzések listázása

## Kölcsönzés

Nem véglegesített kölcsönzések

Aktuális kölcsönzések

Összes kölcsönzés listázása

Azon.	Kölcsönző	Autó azon.	Típus	Időtartam	Kérés ideje	Ár(Ft)	Véglegesít	Törlés
19	andi	2	Renault Clio 1.2	2009-11-07 2009-11-21	2009-04-20 11:44:29	77000	<input checked="" type="checkbox"/>	<input type="checkbox"/>
20	andi	2	Renault Clio 1.2	2009-11-12 2009-11-27	2009-04-21 00:46:34	82500	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Nem véglegesített kölcsönzések listázása

A Kölcsönzés menüben 3 linket találunk : „Nem véglegesített kölcsönzések”, „Aktuális kölcsönzések” és „Összes kölcsönzés listázása”. Mindegyik esetében egy táblázat jelenik meg előttünk. Ezeknek soraiban minden a kölcsönzésekhez szükséges adatot megtalálhatunk , ilyen a kölcsönző felhasználóneve, a kölcsönzendő járműt azonosító car\_id., a kölcsönzési időintervallum, a kölcsönzés felvitelének ideje, a kölcsönzés által fizetendő összeg. A még nem véglegesített kölcsönzések listázása esetén a rootnak lehetősége van a kölcsönzés törlésére illetve véglegesítésére a checkboxok kipipálásával. Erre azért van szükség, mert ha csak 1 lépéses lenne a kölcsönzés, azaz a rootnak nem kellene azt véglegesítenie, akkor az szinkronizációs problémákhoz vezethetne. Gondoljunk csak bele, hogy ha valaki kikölcsönözné online az egyik járművet, és ugyanazt a járművet személyesen kikölcsönözné valaki ugyanarra az időpontra ami alatt a root nem nézte meg az online kölcsönzéseket, akkor nagy problémák lehetnének ebből kifolyólag.

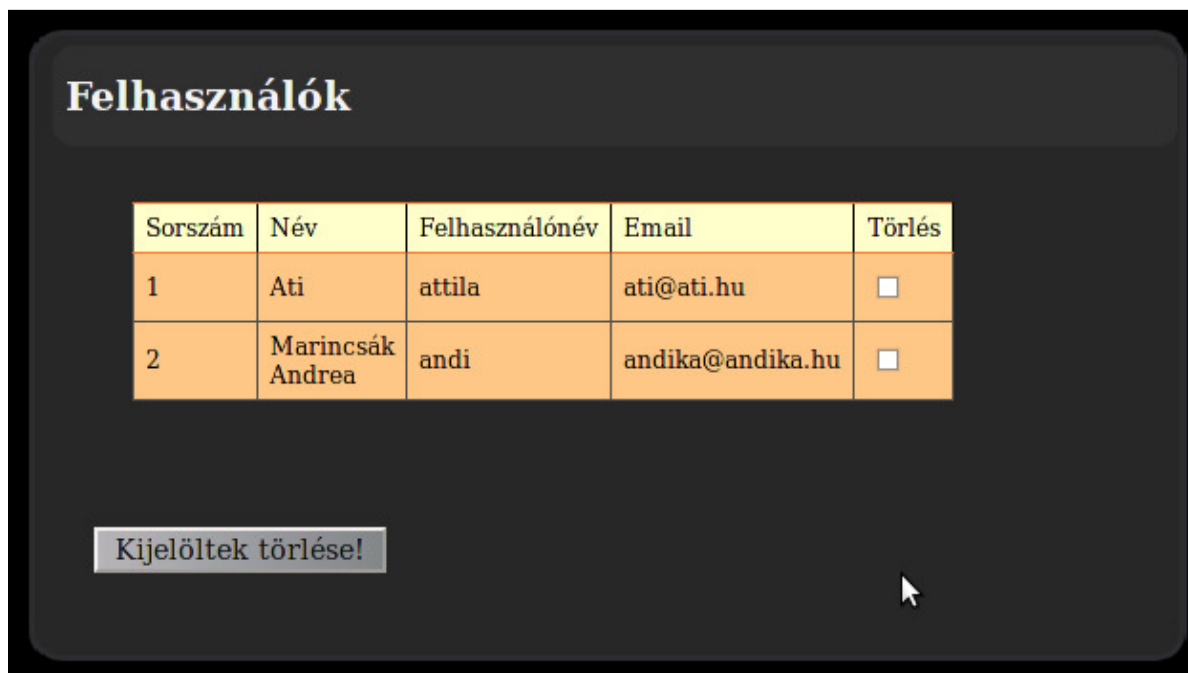
Ha a nem véglegesített kölcsönzéseket szeretnénk listázni , akkor a ListaNoSubmit osztály noSubmittedRent() metódusa hívódik meg, mely létrehozza a táblázatot és ha véglegesítésre kerül sor, akkor meghívja a veglegesitKolcsonzes és rentEmail metódusokat. A rentEmail metódus a már tárgyalt email metódushoz hasonlóan küld egy emailt, melyben tájékoztatást kap a kölcsönzési kérésének elfogadásáról, véglegesítéséről. A veglegesitKolcsonzes és a torolKolcsonzes függvények beállítják a KOLCSONZESEK táblát a megadott utasításnak megfelelően.

Az aktuális kölcsönzések listázása esetén a ListaActual alosztályom actualRent() metódusa fut le, amely csak azokat a kölcsönzéseket listázza ki ,amelyek az aktuális dátum ideje alatt futnak, azaz azon gépjárművekhez tartozó kölcsönzéseket amelyek jelenleg nincsenek a cégnél, hanem ki vannak kölcsönözve. Ez a funkció sok kölcsönzés esetén nagyon hasznos, mert akkor tisztán látja a root, hogy mely gépjárművek nincsenek jelenleg nála.

Az összes kölcsönzés listázásánál a ListaAll alosztály allRent() metódusa fut le. Ez annyiban különbözik az eddigi kölcsönzési táblázatoktól, hogy az összes eddigi kölcsönzést kilistázza és mindegyiknél van egy checkbox, mely a kölcsönzés kifizetéseket tartja karban. Ha ki van fizetve a kölcsönzés akkor a checkbox ki van pipálva és disable tulajdonsággal van ellátva, azaz nem módosítható. Ha viszont nincs kifizetve, akkor mikor kifizeti a kölcsönző személy a kölcsönzés árát, akkor itt kipipálva elmentheti a root a kifizetést.

## Felhasználók listázása

A Felhasználók menüpontot választva szintén egy táblázattal találkozunk. Itt az összes - rooton kívüli- regisztrált felhasználó felhasználóneve és email címe megtalálható. Ezen kívül a rootnak lehetősége van egy személy törlésére is a mellette levő checkbox kipipálásával.



Sorszám	Név	Felhasználónév	Email	Törlés
1	Ati	attila	ati@ati.hu	<input type="checkbox"/>
2	Marincsák Andrea	andi	andika@andika.hu	<input type="checkbox"/>

Kijelöltek törlése!

Felhasználók listázása

## Kinézet

A kinézet egy az Internetről letöltött html+css fájlra épül (a pontos forrás a Források címszó alatt), de pár dolgot módosítottam rajta. Eredetileg nem volt középre igazítva az oldal, és jobb oldalt nem volt egy –most a Bejelentkezésnek és Profilnak helyet adó- box. Az inputok háttérfehér volt, de jobbnak láttam módosítani, így egy szürke árnyalatú képet tettem be háttérnek, amit Gimp-pel szerkesztettem. A táblázatok és szeletek kinézetét is én csináltam. A fekete, szürke, narancs, fehér színek és ezek árnyalatai dominálnak az oldalon.

Az oldal kinézete alapvetően 6 részből áll, ahol az első a header, a második a navlist-container, amiben az oldal felső részében található menüpontok vannak, a harmadik az a content, ami a bal oldalon található boxokat tartalmazza, a negyedik pedig a post-container,

ami középen található, ahol megjelennek a főbb funkciói az oldalnak, az ötödik a jobb oldalon található boxot tartalmazza, ami a bejelentkezésért és a profilért felelős, ez a content-belepes , és végül a footer, ami a lap alján található .

## **IV. Összefoglalás**

A program, az elkészítése előtt eltervezett, kigondolt funkciókat teljes mértékig véghez tudja vinni. Lehet regisztrálni, profilt módosítani, gépjárműveket listázni, kölcsönözni, a rootnak pedig lehetősége van a gépjárművek szerkesztésére, új gépjármű hozzáadására, a felhasználók és a kölcsönzések karbantartására.

A program kinézete bár látványos, de egy komolyabb autókölcsönző weboldalának a kinézetéhez azért nem hasonlítható. Az oldal Mozilla Firefox böngésző alá optimalizált. A kód érthető, áttekinthető, de így utólag már egy-két dolgot másképp valósítanék meg, illetve a program felépítését is módosítanám, egy meglévő sablonra –mint pl. a bevezetésben említett Zendre- támaszkodnék.

A fejlesztés ideje –kezdő PHP fejlesztő lévén- elég hosszadalmas volt, de ez idő alatt sok új dolgot tanultam. A programban több helyen megfigyelhető, hogy 2 ugyanolyan részfeladatot másképp oldottam meg, ez pont a sok új, praktikusabb dolgok felfedezésének az eredménye. Próbáltam több olyan funkciót alkalmazni a program során, ami érdekesebbé teszi, ilyen például a 2 lépéses regisztráció és a kölcsönzés véglegesítése esetében az automatikus email küldés, a kölcsönzési időintervallum meghatározásánál a jquery használata, vagy például az FCKEditor program használata az egyes oldalak szerkesztésénél.

Összességében véve egy érdekes alkalmazás, mely a teljesség igénye nélkül kíván eleget tenni egy autókölcsönzéssel foglalkozó cég követelményeinek.

Így utólag is örülök, hogy ezt a témát választottam, így elértem célomat, azaz, hogy jobban megismerkedjek a PHP nyelvvel. Várakozásaimnak megfelelően nem csalódtam a programnyelvben, nagyon érdekesnek találom és még jobban meg szeretnék ismerni a továbbiakban.

## **V. Felhasznált technológiák, alkalmazások**

APACHE 2.2.9 (Linux)

MYSQL 5.0.67

PHP 5.2.6

PHPMYADMIN 2.11.8.1deb1

GIMP 2.6.1

## **VI. Irodalomjegyzék**

Peter Moulding : PHP Fekete könyv

### **Felhasznált Internetes adat:**

1. A PHP nyelv hivatalos dokumentációja: <http://hu.php.net/manual/en>
2. <http://www.w3schools.com/>
3. <http://hu.wikipedia.org/wiki/PHP>
4. <http://carthemes.net/>

## **VII. Függelék**

A README.txt fájl tartalmazza a program összes funkciójának futásához szükséges beállításokat.