

Adatbázis kezelés PHP környezetben

Témavezető:
Dr. Boda István
egyetemi docens

Készítette:
Széles Szabolcs
programtervező matematikus

Tartalomjegyzék

1. Bevezető	4
2. Szabad és nyílt forráskódú szoftverek	5
2.1 Szabad szoftver	5
2.2 Nyílt forráskódú szoftver	6
2.3 Elterjedtség	6
2.4 Webes alkalmazások	7
2.4.1 Statikus weboldalak	7
2.4.2 Dinamikus weboldalak	8
2.5 Dinamikus weboldalak készítése PHP, PostgreSQL felhasználásával	9
3. Az alkalmazás fejlesztésének menete	10
3.1 Funkcionális követelmények	10
3.2 Nem funkcionális követelmények	12
3.3. Komponens elemzés	13
3.4 Adatbázis megtervezése	14
3.4.1 A táblák szerkezete	14
3.4.2 Adatbázis interfész	20
3.4.3 Adatkapcsolati diagram	21
3.4.4 Biztonsági mentések készítése	22
3.5 Különböző felhasználók által elérhető funkciók	22
3.6 Validáció	23
4. A webáruház felépítése a boltosok szemszögéből	24
4.1 Telepítés	24

4.2 Az alkalmazás használata	24
4.3 Eltávolítás.....	37
5. A webáruház felépítése a vásárlók szemszögéből.....	38
6. Összegzés.....	42
7. Felhasznált irodalom	43

1. Bevezető

A dolgozatomban egy webáruházon keresztül próbálom bemutatni, hogy milyen adatbázis kezelési lehetőségek vannak PHP-ben, illetve a webáruház fejlesztésének folyamatát mutatom be, majd a felhasználók számára szükséges információkat. Ehhez a PostgreSQL adatbázis kezelő rendszert választottam mind tudása, mind szabad szoftver volta miatt. A választásom azért pont egy webáruházra esett, mivel a különböző függvények bemutatását egyrészt nagyon száraznak találtam, másrészt ezeknek a leírása megtalálható a gyártók oldalain, bár tény, hogy többnyire angolul. Így egy valamennyire látványos formát öltött ennek a témának az ismertetése. A munkám során nyílt forráskódú programokkal dolgoztam.

Itt kívánok köszönetet mondani mindazoknak, akik munkámat segítették, javaslataikkal vagy éppen kritikáikkal előre lendítették azt.

2. Szabad és nyílt forráskódú szoftverek

2.1 Szabad szoftver

A közkincként (public domain) terjesztett forráskód egyidős a modern programozással. A szabad szoftver mozgalmat Richard M. Stallman indította 1983-ban a GNU projekttel. Az elvi célkitűzést kiáltvány (The GNU Manifesto, 1984), alapítvány (Free Software Foundation, röviden FSF, 1985) és az első általános szabad szoftver licenc (GNU GPL, 1989) követte. Az FSF szerint a szabad szoftverek a következő szabadságjogokkal kell, hogy rendelkezzenek:

1. A tetszőleges célra történő szabad felhasználás;
2. a szabad tanulmányozhatóság és igény szerinti módosíthatóság, aminek előfeltétele a forráskódhoz való hozzáférhetőség;
3. a másolatok szabad terjeszthetősége, segítve ezzel ismerőseinket;
4. a szabad továbbfejleszthetőség, és az eredmény szabad közzététele a közösség javára. Ennek is előfeltétele a forráskód elérhetősége.

A „szabad” nem feltétlenül jelent „ingyenes”: bárki bármennyiért árusíthatja a kérdéses programokat; az egyetlen feltétel, hogy a fenti négy alapjogot garantálja vevői számára. Miért venné meg bárki is? Azért, mert például nem képes azt magának lefordítani, szüksége van kézikönyvre, CD-n vagy DVD-n szeretné a programokat megkapni, vagy mert támogatásra van szüksége. Az is elképzelhető, hogy valaki egyedi fejlesztést, testreszabást, adott hiba kijavítását, adott funkció beépítésével bíz meg egy programozót vagy céget.

Az FSF vezeti a fenti elveknek megfelelő licencek listáját.

A GNU GPL a legelterjedtebb szabad szoftver licenc. A nyílt forráskódú fejlesztések nagy része pedig az FSF vezetésével készült nyílt forráskódú fejlesztőeszközöket használja mind a mai napig.

2.2 Nyílt forráskódú szoftver

Eric S. Raymond és Bruce Perens kezdeményezésére, a szabad szoftverek vállalati körben való népszerűsítésére 1998-ban megalapult az Open Source Initiative (OSI) közhasznú társaság, amely a nyílt forráskódot hangsúlyozza az angolban félreérthető „szabad” helyett. (A szabad szoftver angol nevében szereplő free ingyenes jelentéssel is bír, ami tévesen azt sugallhatja, hogy szabad szoftvereket nem lehet eladásra fejleszteni, vagy egyéb profitorientált módon felhasználni.)

A nyílt forráskódú licenceket az OSI véleményezi és tartja nyilván. A nyílt forráskódú licencek között megtalálni a FSF GNU licenceit, de a Microsoft Ms-PL és Ms-RL licenceket is.

2.3 Elterjedtség

Az internetet és a világhálót nyílt forráskódú szoftverek hajtják: a BIND névkiszolgáló, az Apache webkiszolgáló és a Sendmail levélszerver mind piacvezető a maga nemében. A világ 500 legnagyobb számítási teljesítményű szuperszámítógépének (TOP500) 85%-a Linux operációs rendszert futtat. A webes keresők piacvezetője, a Google Linuxra és más nyílt forráskódú alkalmazásokra építette piaci sikerét. Az Oracle az Oracle Unbreakable Linux támogatási programmal és az Oracle Enterprise Linux terjesztésével nyílt forráskódú alapokra helyezi piacvezető adatbázis-kezelőjét. A címtár- és személyazonosság-felügyeleti termékek piacvezető vállalata, a Novell a SUSE Linux felvásárlásával és erre épülő új termékeivel a nyílt forráskód egyik legfőbb propagálójává és támogatója lépett elő. A Sun Microsystems az OpenOffice.org után piacvezető hardvereinek operációs rendszerét, a Solarist, valamint Java fejlesztőeszközeinek legfontosabb elemeit is nyílt forráskódúvá tette. A Microsoft, a piaci érték alapján a világ legnagyobb szoftvercége több termékét, például az Ajax Control Toolkit, az IronPython vagy az XML Notepad 2007

programokat tette nyílt forráskódúvá a Microsoft közösségi nyílt forráskódú fejlesztőoldalán, a CodePlexen.¹

2.4 Webes alkalmazások

Az Internetet böngészve alapvetően két fajta web-oldallal találkozhatunk szembe: a statikus oldalakkal, melyeket akárhányszor is töltünk újra mindig pontosan ugyanazt az adattartalmat adják vissza és semmilyen interakciót nem biztosítanak számunkra. Ugyanakkor számos olyan oldal is fellelhető, amelyek tartalma dinamikusan változik, azaz műveleteinktől, a működési környezettől valamint egyéb tényezőktől függően több vagy kevesebb, de változó információt (is) tartalmaznak.²

2.4.1 Statikus weboldalak

Az előnyök közé sorolható, hogy az ilyen oldalak könnyen és gyorsan összeállíthatóak még az ezen területen kevés tapasztalattal felhasználók számára is. Alkalmas nem, vagy ritkán változó információk közlésére, valamint arra, hogy megmutassuk, egy későbbi dinamikus weboldal hogyan fog kinézni.

Hátránya, hogy mivel kézzel kell frissíteni az oldal forráskódját, így nagy mennyiségű adatnál ez elég nehézkesé válhat. A naprakészséget emiatt nehéz megoldani, valamint a felhasználók semmiféle testre szabhatóságot nem kapnak.

¹ <http://prog.hu/cikkek/180/Dinamikus+weboldalak.html>

² <http://prog.hu/cikkek/180/Dinamikus+weboldalak.html>

2.4.2 Dinamikus weboldalak

A dinamikus weboldalakot csoportosíthatjuk.

- szerver-oldali alkalmazás
- kliens-oldali alkalmazás
- vegyes alkalmazás

Szerver oldali alkalmazások közé sorolhatóak a CGI programok, melyek valamilyen programozási nyelv segítségével állítunk elő (C, JAVA), de lehetőség van egyéb lehetőségek, pl. PERL, Shell scriptek, kötegelt állományok felhasználására is. Ezeket a segédprogramokat az ügyfél böngészőprogramja indítja el, számára láthatatlanul futnak a szerveren és csak dolguk elvégzése után jeleznek vissza.

Manapság már előtérbe került-szerver oldali alkalmazások közé sorolható a PHP, ASP, JSP. Ezeknél a módszereknél a böngésző egy kész HTML állományt kap, melyet minden böngésző meg tud jeleníteni, az adatok feldolgozását pedig a szerver végzi.

A kliens oldali alkalmazások futtatását a böngésző motorja végzi, ám a különböző böngészők képességei jelentősen eltérhetnek egymástól és itt nem kizárólag a sebességre kell gondolni, hanem a kód értelmezésére is. Ez jelentősen hozzájárul az implementációs hibák számának növeléséhez, hiszen ugyanazt az oldalt két különböző böngészőn akár teljesen máshogy is láthatjuk, s ritkább esetben még akár az is előfordulhat, hogy egy régebbi böngésző motorja az alkalmazás egy újabb változatát még futtatni sem képes. A kliens oldali alkalmazások közé tartozik a Javascript, VBScript.

Biztonsági szempontból sajnálatos dolog, hogy a kliens oldali alkalmazásoknál a felhasználó hozzáférhet a kódhoz és meg is változtathatja azt, így emiatt semmilyen, az adatok biztonságát veszélyeztető funkciót nem láthatnak el.

Amennyiben a két módszert kombináljuk, úgy egy kevert megoldást kapunk. A kliens oldali alkalmazással csökkenthető a szerver terhelése. A megvalósítás során a Javascriptet kizárólag űrlapmezők adatainak ellenőrzésére használom, hogy a mezők egyáltalán ki vannak-e töltve, ám a jelszavak helyességének ellenőrzése már minden esetben szerver oldalon történik.

2.5 Dinamikus weboldalak készítése PHP, PostgreSQL felhasználásával

A dinamikus weboldalak készítése során gyakran használunk valamilyen adatbázist az adatok rendezett tárolása céljából. Igényeinknek és lehetőségeinknek megfelelően nagyon sok szoftver közül választhatunk, melyek tudásukban jelentősen eltérnek egymástól.

A PostgreSQL egy ingyenes felhasználható, BSD licensszel rendelkező, ám igen komoly tudással felvértezett adatbázis kezelő rendszer. Bonyolultabb feladatok ellátására is képes. Ismeri a tárolt eljárásokat, triggereket, beágyazott típusokat is képes kezelni. A szoftver eredeti neve Postgres volt. Amikor SQL funkcionalitással egészítették ki 1995-ben, a nevét Postgres95-re változtatták. 1996 végén kapta mai nevét.³

A PHP-t szerver-oldali weboldalak generálására használják. 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.⁴

Számtalan adatbázis kezelő rendszert ismer a kiegészítései révén. Gyengén típusos nyelv, az objektum orientált programozásra is lehetőséget ad, szintaktikája C-szerű.

Ezen kívül a két programon kívül szükség van még egy webserverre is. Erre a célra az Apache HTTP Server-t választottam, mely egy igen jól konfigurálható,

³ http://www.postgresql.org/docs/faqs.FAQ_hungarian.html#1.2

⁴ <http://hu.wikipedia.org/wiki/PHP>

szintén nyílt forráskódú alkalmazás, mely több operációs rendszer alatt is elérhető. A névválasztás oka kettős. Egyrészt az állóképességükről híres apacs indiánoknak tiszteleg, akik az Egyesült Államok dél-nyugati részén élő őslakók. Másrészt pedig egy szójáték: „a patchy server”, ami arra utal, hogy a szervert eleinte a webmesterek által készített patchekkel frissítették. Az utóbbi azonban inkább egy szerencsés egybeesés eredménye.⁵

3. Az alkalmazás fejlesztésének menete

3.1. Funkcionális követelmények

1. bejelentkezés

Bejelentkezésre amiatt van szükség, hogy pontosan tudjuk, hogy ki vásárolt.

2. felhasználók regisztrálása, törlése, adatmódosítás

A regisztráció során a felhasználók lényeges adataikat megadhatják, melyeket a későbbiekben módosítani is lehet.

3. felhasználók kategorizálása

A felhasználókat több csoportra kell bontani, így elkülönítve a különböző jogköröket.

4. terméktípusok létrehozása, módosítása, törlése

A termékeket különböző kategóriákba lehessen besorolni, valamint biztosítani kell a lehetőséget ezeknek módosítására.

5. szállítmányok felvitele

⁵ http://hu.wikipedia.org/wiki/Apache_HTTP_Server

Biztosítani kell, hogy a megvásárolt termékeket szállítmányok formájában tároljuk, ezzel igazolva a származásukat.

6. a termékek egyediségének biztosítása

Minden egyes termék életútját pontosan nyomon lehessen követni.

7. felhasználók a korábbi rendelkezéseiket megtekinthessék

A korábbi rendelkezések megtekintésére és azok pontos részleteinek megtekintésére lehetőséget kell adni.

8. biztonságos vásárlás feltételeinek megteremtése

Biztosítani kell, hogy az ügyfél által megvásárolt termék valóban az ő nevében legyen megvásárolva, illetve valóban az a termék kerüljön be a kosárba, amit a felhasználó kiválasztott.

9. az adatbázisról biztonsági mentés készítése

Nagyon fontos követelmény, hogy bármilyen szoftveres vagy hardveres hiba miatt fellépő adatvesztés után a teljes adatbázis visszaállítható legyen egy korábbi ép állapotába.

10. hibatűrés

A felhasználói oldalon történő hibák folytán a kosár tartalma maradjon meg, hogy ne kelljen újra összeválogatni a termékeket.

3.2 Nem funkcionális követelmények

1. használhatóság

A felhasználói felület legyen egyszerű, könnyen átlátható, lehetőleg a minimális meglepetés elvének szem előtt tartásával, hiszen elsősorban nem szakemberek fognak a boltban vásárolni, így ha túlbonyolítjuk az oldal szerkezetét, az negatívan befolyásolhatja a vevő vásárlói szándékát.

2. megbízhatóság

A vevőnek látnia kell, hogyha valamilyen hiba folytán egy általa végrehajtott utasítást a rendszer képtelen volt feldolgozni, illetve nyugtázni kell, ha megfelelően lezárult egy tranzakció.

3. hordozhatóság

A rendszer legyen platform független, hogy különböző operációs rendszerek alatt is ugyanúgy működjön.

4. implementációs követelmény

Mivel a webáruház weben működik, így olyan HTML, CSS, Javascript kódot használjunk, ami (majdnem) minden böngészőn ugyanazt az eredményt szolgáltatja.

5. adatvédelem

Biztosítani kell, hogy mindenki csak a számára szükséges adatokat láthassa, máshoz ne férhessen hozzá.

6. biztonság

A bejelentkezéssel biztosítjuk, hogy a felhasználók önmagukat azonosítva csak a saját nevükben vásárolhassanak. Természetesen a jelszavak mások által való megszerzéséért a boltot nem terheli felelősség.

3.3 Komponens elemzés

1. PostgreSQL

A PostgreSQL a megfelelő eszközt biztosítja az adatok tárolásához, lekérdezéséhez, módosításához.

2. PHP

A PHP 5-ös biztosítja az adatbázis elérését, valamint a felhasználó kezelést munkamenet azonosítón keresztül.

3. Apache

Az Apache webservert biztosítja a PHP futását, mely a HTML kódot előállítja.

4. TCPDF

Nyílt forráskódú PHP nyelven írt program, amely PDF fájlokat képes előállítani.

3.4 Adatbázis megtervezése

3.4.1 A táblák szerkezete

1. *USERS* tábla

Mező	Típus	Null	Alapértelmezett
*id	integer	Nem	auto increment
firstname	varchar(20)	Nem	
lastname	varchar(20)	Nem	
phone_number	varchar(11)	Nem	
email_address	varchar(40)	Nem	
login_name	varchar(16)	Nem	
user_password	varchar(32)	Nem	
address	t_address	Nem	
active	boolean	Nem	false
firm_name	varchar(128)	Nem	'' :character varying
registered	timestamp	Nem	now()
hash	varchar(16)	Nem	
admin	boolean	Nem	false

Magyarázat:

*id	felhasználó azonosító
firstname	keresztnév
lastname	vezetéknév
phone_number	telefonszám
email_address	e-mail cím
login_name	login név
user_password	felhasználó jelszavának md5 hash alakja
address	cím
active	aktív-e a felhasználó
firm_name	cégnév (Nem kötelező megadni, amennyiben magánszemély.)
registered	regisztráció időpontja
hash	a megerősítéshez szükséges hash kód
admin	kitüntetett felhasználó-e

Ebben a táblában tárolom a felhasználók adatait. Ebbe beletartoznak a vásárlók, illetve a bolt üzemeltetői is. A felhasználó azonosítására egy mező van kitüntetve, melyet az adatbázis kezelő automatikusan rendel a személyhez. A cím egy saját, összetett típus. A *HASH* mező adatbiztonsági okok miatt szükséges. A regisztráció során egy megerősítő e-mailt küldök ki, melyben egy hivatkozásra

kattintva megerősítheti a felhasználó, hogy igen, ő akart regisztrálni, ez az ő e-mail címe. Az *ADMIN* mező pedig azt jelöli, hogy bolti dolgozóról van-e szó.

2. *ORDERS* tábla

Mező	Típus	Null	Alapértelmezett
*order_id	integer	Nem	auto increment
user_id	integer	Nem	
status_comment	varchar(128)	Nem	'-'
order_list	t_order_list[]	Nem	
status	t_status	Nem	'ordered'::t_status
order_time	timestamp	Nem	now()

Magyarázat:

*order_id	rendelés azonosító
user_id	felhasználó azonosító
status_comment	megjegyzés a kiszállításhoz
order_list	rendelések listája
status	rendelés állapota
order_time	a rendelés leadásának ideje

Ebben a táblában tárolom a leadott megrendeléseket. A rendelés azonosítóját az adatbázis kezelő rendeli hozzá a leadott rendeléshez. A felhasználó azonosítója egy létező felhasználó azonosítója a *USERS* táblából, ezzel egyértelműen azonosítható, hogy ki adta le a megrendelést. A rendelések listája egy saját összetett típus tömbje, melynek egy-egy mezőjében egy-egy termék kerül tárolásra azoknak a leadás pillanatában rendelkező jellemzőivel.

3. *PRODUCTS* tábla

Mező	Típus	Null	Alapértelmezett
*id	integer	Nem	auto increment
supply_id	integer	Nem	
status	varchar(32)	Nem	'available'::character varying
status_comment	varchar(128)	Igen	
name	varchar(32)	Nem	

Magyarázat:

*id	termékazonosító
supply_id	szállítmány azonosító

status termék aktuális állapota
status_comment állapothoz megjegyzés
name termék neve

Ebben a táblában tárolom, hogy milyen konkrét termékek vannak raktáron. Minden egyes termék saját azonosítóval rendelkezik, melyet az adatbázis kezelő rendel hozzá a termékhez és melynek segítségével a későbbiekben a termék egyértelműen beazonosítható. A szállítmány azonosító egy konkrét szállítmány azonosítója a *SUPPLIES* táblában, melynek segítségével nyomon követhető, hogy az adott terméket mikor kitől vette át a bolt. A termék állapotát akkor változtatom meg, ha valaki a kosárba teszi, hogy egyszerre két kosárba ne kerülhessen be.

4. *SUPPLIES* tábla

Mező	Típus	Null	Alapértelmezett
*id	integer	Nem	auto increment
waybill	varchar(20)	Nem	
date	date	Nem	
partner_id	integer	Nem	
products	t_supply_list	Nem	

Magyarázat:

*id szállítmány azonosító
waybill fuvarlevél száma
date átvétel dátuma
partner_id szállító azonosító
products szállítmányban lévő termékek listája

Ebben a táblában tárolom a beérkező szállítmányokat. Az azonosítót ebben az esetben is az adatbázis kezelő rendeli a szállítmányhoz. A fuvarlevél számával és a partner azonosítójával pedig egyértelműen meg tudom határozni, hogy honnan származik az áru. A *PRODUCTS* mező egy saját összetett típus, melyben tárolom, hogy melyik termékből hány darab érkezett és milyen áron.

5. PRODUCT_LIST

Mező	Típus	Null	Alapértelmezett
*id	integer	Nem	auto increment
name	varchar(32)	Nem	
category	integer	Nem	
net_price	double precision	Nem	0.0
vat	double precision	Nem	0.2
description	varchar(256)	Nem	'' :character varying
gross_price	double precision	Nem	0.0
vendor_id	integer	Nem	0
link	varchar(128)	Nem	'http://' :character varying
photo	varchar(128)	Nem	'images/default_picture.png' :character varying
deleted	boolean	Nem	false

Magyarázat:

*id	termékazonosító
name	termék neve
category	kategória azonosító
net_price	nettó ár
vat	áfa
description	termékleírás
gross_price	bruttó ár
vendor_id	gyártó azonosító
link	termék honlapjának a címe
photo	fénykép a termékről
deleted	logikailag törölt-e

Ebben a táblában az egyes termékcsoportokat tárolom, nem konkrét termékeket, viszont az egyes termékcsoportok nevének egyezni kell a konkrét termékek nevével. A termékcsoportok azonosítóját az adatbázis kezelő rendeli a termékcsoportokhoz. A termékcsoportok minden egyes jellemzőjét tárolom, melyhez a vásárlásnál szükség lehet, valamint egy logikai törlést is biztosítok, amennyiben a terméket valamilyen oknál fogva nem szeretném, hogy megjelenjen a kínálatban.

6. CATEGORIES

Mező	Típus	Null	Alapértelmezett
*id	integer	Nem	auto increment
name	varchar(32)	Nem	
parent	integer	Nem	-1
level	integer	Nem	

Magyarázat:

*id	kategória azonosító
name	kategória neve
parent	szülőkategória
level	kategória szintje

Ebben a táblában tárolom a különböző kategóriákat, melyekbe a termékeket besorolom. A 3 szintig lehet az altípusokat megadni, tehát összesen 4 kategóriaszint megadására van lehetőség minden egyes terméknel, mely véleményem szerint elegendő. Az egyes kategóriaszinteken nem lehetnek egyező nevű alkategóriák ugyanazon szülő esetén, mivel ennek értelme sem lenne.

7. VENDORS

Mező	Típus	Null	Alapértelmezett
*id	integer	Nem	auto increment
name	varchar(128)	Nem	
web_address	varchar(256)	Nem	'http://':character varying

Magyarázat:

*id	gyártó azonosító
name	gyártó neve
web_address	gyártó weboldala

Ebben a táblában a különböző gyártók adatait tárolom, ám csak a nevét és a honlapjának a címét tartottam fontosnak eltárolni.

8. PARTNERS

Mező	Típus	Null	Alapértelmezett
*id	integer	Nem	auto increment
name	varchar(128)	Nem	
address	t_address	Nem	
tax_invoice	varchar(11)	Nem	
phone_number	varchar(11)	Nem	
email_address	varchar(32)	Nem	
still_partner	boolean	Nem	true

Magyarázat:

*id	partner azonosító
name	partner/cég neve
address	cím
tax_invoice	adószám
phone_number	telefonszám
email_address	email cím
still_partner	még mindig partner-e

A táblában a nagykereskedelmi egységek lényeges adatait tárolom, mely a velük való kapcsolattartásban szerepet játszik, illetve melynek alapján nyilván tudom tartani, hogy a bolt pontosan kitől vette a termékeket.

9. CARTS

Mező	Típus	Null	Alapértelmezett
*id	integer	Nem	auto increment
userid	integer	Nem	
product_list	t_product_list[]	Nem	

*id	kosár azonosító
userid	a felhasználó azonosítója
product_list	a kosárban lévő termékek listája

Ez a tábla a vevők által kosárba helyezett termékeit tárolja azoknak azonosítójával és mennyiségével.

3.4.2 Adatbázis interfész

Az adatbázis kezelési műveleteket megkönnyítendő, illetve a későbbiekben másik adatbázis kezelőre való áttérést segítő egy adatbázis interfészt hoztam létre. Az osztály neve *DBConnection*.

Az adatbázissal való kapcsolat felvételéhez szükséges felhasználó név, jelszó, az adatbázis elérési útja, illetve az adatbázis neve egy külön fájlban került eltárolásra *DBConfig.php* néven.

Az általam készített metódusok elfedik a valódi PHP által biztosított függvények neveit, azoknak használati módját. Erre egy jó példa az adatbázishoz való kapcsolódás.

Ahelyett, hogy a php által biztosított módon

```
$connection = pg_pconnect("dbname=online_store
user=felhasznalo password=jelszo host=localhost");
if (!$connection)
{
print("Connection Failed.");
exit;
}
```

kapcsolódnánk az adatbázishoz, ahelyett elegendő, ha az osztály használatával a

```
$connection= new DBconnection();
```

kódot írjuk be.

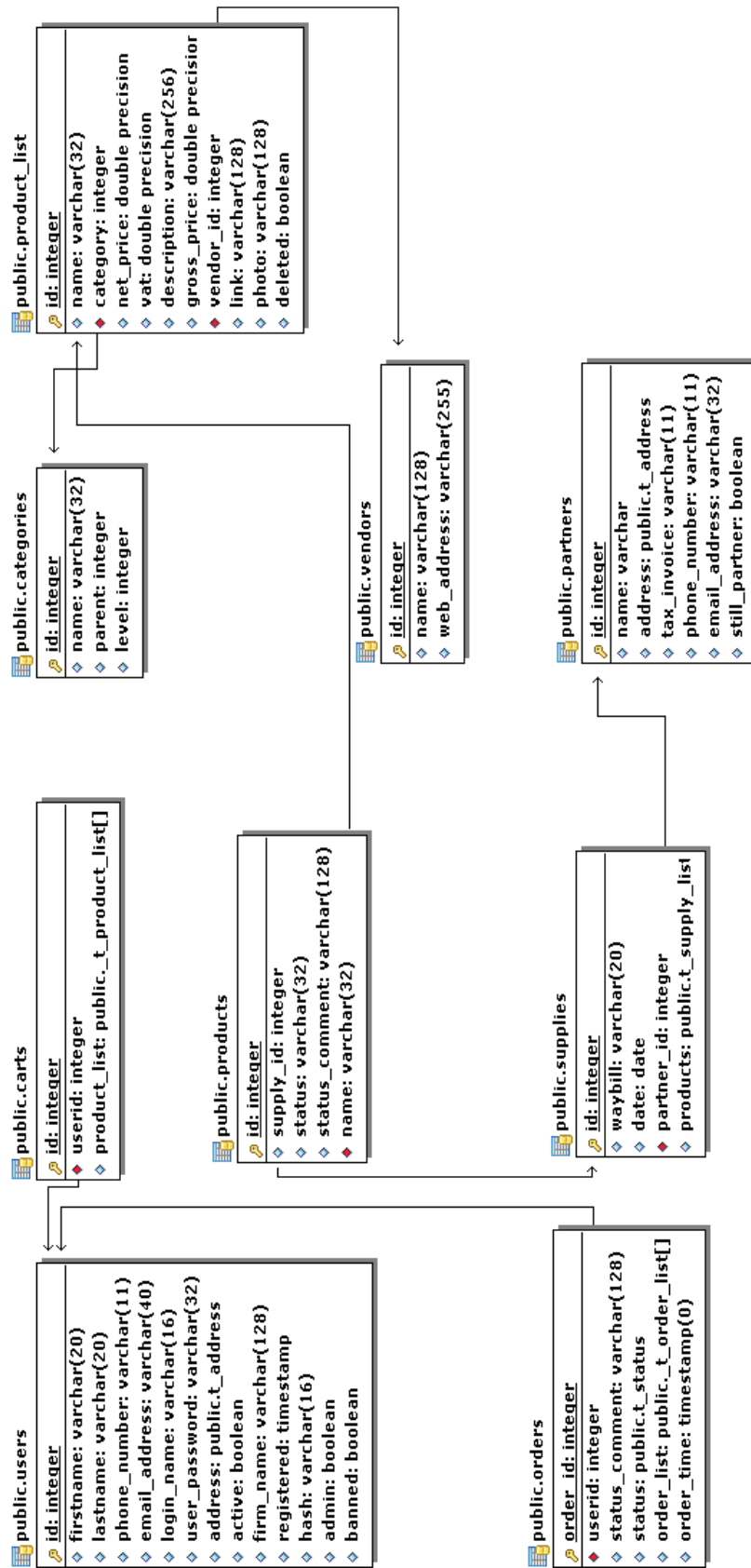
Lehetőségünk van a kapcsolat általi utolsó lekérdezés eredményének megtekintésére az adatbázisból való újbóli lekérdezés nélkül, vagy az utoljára módosított sorok számának lekérdezésére, vagy annak megtekintésére, hogy a lekérdezésben hány sor szerepel.

Például a

```
$connection->getUpdatedRows();
```

visszaadja, hány sort módosított az előző UPDATE művelet.

3.4.3 Adatkapcsolati diagram



3.4.4 Biztonsági mentések készítése

Biztonsági mentés készítése egyrészt implementálva van, tehát ha ezt a lehetőséget választjuk, akkor egy megfelelően paraméterezett kód elindítja a PostgreSQL rendszer biztonsági mentés funkcióját és ezután az adatbázis törlésével, majd az adatbázis újbóli létrehozásával a fájlba mentett kód futtatásával visszaállíthatjuk az előző állapotot. Természetesen lehetőség van bármilyen más külső program használatára is, mely képes biztonsági mentést készíteni az adatbázisról.

```
$exec_string="\c:\\program
files\\postgresql\\8.3\\bin\\pg_dump.exe\" -U postgres
online_store
>database_backup\\online_store_\".($date=(date("Ymd_His
"))).\".sql\";
$result=exec($exec_string,$output);
```

Célszerű akkor készíteni a biztonsági mentést, amikor kevés a felhasználó. Erre a legcélravezetőbb mód az, ha valamikor hajnalra egy időzített programfuttatást ütemezünk.

3.5 Különböző felhasználók által elérhető funkciók

Boltosok

1. Új boltos jogkörrel felruházott felhasználó regisztrálása, illetve annak törlése, módosítása.
2. Felhasználó letiltása illetve engedélyezése
3. Felhasználók keresése, listázása
4. Új kategóriák létrehozása, illetve azoknak törlése
5. Termékek kategóriákba sorolása, illetve azoknak módosítása, törlése
6. Új termékbejegyzés létrehozása, azoknak módosítása

7. Rendelések állapotának lekérdezése, módosítása
8. Új partner létrehozása, módosítása
9. Új gyártó létrehozása, módosítása
10. Biztonsági mentés készítése az adatbázisról.

Regisztrált felhasználók

1. Bejelentkezés.
2. Saját adatok módosítása a felhasználói név kivételével.
3. Termékek kosárba helyezése.
4. A kosár tartalmának törlése.
5. Termékek megvásárlása.
6. Korábbi rendelések megtekintése.
7. Termékek adatainak megtekintése.

Nem regisztrált felhasználók

1. Regisztráció
2. Termékek adatainak lekérdezése.

3.6 Validáció

A validáció arra lenne hivatott, hogy megmondja, a megrendelő igényeinek megfelelő terméket hoztam-e létre. Mivel ez esetben én magam voltam a megrendelő és a kivitelező is, így mondhatjuk, hogy az igényeknek majdnem teljes mértékben megfelelő terméket hoztam létre, ám mivel a weboldal kinézetével nekem is problémáim vannak, így annak megtervezését egy ahhoz értő grafikusra bízám.

4. A webáruház felépítése a boltosok szemszögéből

4.1 Telepítés

1. Apache webservertelepítése, beállítása
2. PHP telepítése, beállítása
3. PostgreSQL telepítése, beállítása
4. Adatbázis létrehozása
5. *online_webshop.sql* file lefuttatása, ami létrehozza a megfelelő táblákat
6. A php állományok a beállításoknak megfelelő helyre történő másolása
7. Az adatbázis feltöltése adatokkal

Ennek a folyamatnak a végén célszerű az adatbázisról egy biztonsági másolatot készíteni.

4.2 Az alkalmazás használata

Ezt a felületet kizárólag a boltos jogkörrel rendelkező felhasználók láthatják.

A következő menüpontok illetve funkciók érhetők el az áruház használata során:

1. Bejelentkezés

A nyitóoldalon megjelenő felhasználónév és jelszó párost ki kell tölteni, majd az OK gombra kattintva megtörténik a bejelentkezés, mely során egy munkamenet azonosítóban el lesz tárolva a felhasználó azonosítója és neve. Megkezdhetjük az alkalmazás használatát.

2. Új kategória

Új termékkategória létrehozására ad lehetőséget. Összesen 3 alkategória létrehozására van lehetőség egy főkategóriához. A program felügyeli, hogy egy adott főkategóriában két azonos nevű alkategória ne szerepelhessen

ugyanazon a szinten, ám két különböző főkategóriához tartozó azonos szinten lévő al kategória neve minden további nélkül megegyezhet. A kategória adatbázisban való tárolása egy INSERT művelet segítségével történik.

Bejelentkezve: [szelessz](#)
[Kijelentkezés](#)

Új kategória
Termék másik kategóriába sorolása
Kategória törlése

Új termék
Termékek tulajdonságainak szerkesztése

Új partner
Partner szerkesztése

Új gyártó
Gyártó adatainak módosítása

Új szállítvány
Szállítványok megtekintése

Adatbázis mentése

Leadott rendelések megtekintése

Boltos jogosultság
Felhasználók letiltása, engedélyezése

Új kategória hozzáadása

Első szint: Aleplep ▼

Második szint: Válassz ▼
Válassz
LGA-775
Socket-AM2

Harmadik szint: Válassz ▼

Új kategória neve:

Az adatbázis *CATEGORIES* táblájában kerül tárolásra egy kategória, a megfelelő szintre történő besorolás a PHP kód feladata.

Az egyes kategóriák tartalma legördülő menü formájában jelenik meg, a menük mindig csak az egy szinttel magasabban kiválasztott kategóriához tartozó al kategória tartalmát jelenítik meg dinamikusan Javascript kód segítségével.

3. Termék új kategóriába helyezése

Ebben a menüpontban a kiválasztott terméket lehet egy másik termékkategóriába sorolni. A szintek kiválasztása az új kategória létrehozásánál ismertetett dinamikus PHP kóddal generált Javascript kóddal történik.

The screenshot shows a web application interface with a sidebar on the left and a main content area on the right. The sidebar contains a list of menu items: "Bejelentkezve: szelessz", "Kijelentkezés", "Új kategória", "Termék másik kategóriába sorolása", "Kategória törlése", "Új termék", "Termékek tulajdonságainak szerkesztése", "Új partner", "Partner szerkesztése", "Új gyártó", "Gyártó adatainak módosítása", "Új szállítmány", "Szállítmányok megtekintése", "Adatbázis mentése", "Leadott rendelések megtekintése", "Boltos jogosultság", and "Felhasználók letiltása, engedélyezése". The main content area is titled "Termék másik kategóriába sorolása" and contains a form with four dropdown menus labeled "Első szint:", "Második szint:", "Harmadik szint:", and "Negyedik szint:". The "Első szint:" dropdown is set to "Fő kategória", "Második szint:" to "Alkategória 1", "Harmadik szint:" to "Alkategória 2", and "Negyedik szint:" to "Válassz". Below these is a "Termék neve:" label and a dropdown menu showing a list of products: "Asus M2N-E", "GA-MA770-DS3", "GP", "macska", "Samsung 500Gb SATA", "Samsung Floppy drive", and "Western Digital 250GB SATA2".

```

<script language="javascript">
addOption(document.change_category.level1, "Alaplap",
"Alaplap", "");
addOption(document.change_category.level1, "új2",
"új2", "");
...
function Selectlevel2()
{
removeAllOptions(document.change_category.level2);
addOption(document.change_category.level2, "",
"Válassz", "");
Selectlevel3();
if(document.change_category.level1.value ==
"Alaplap")
{
addOption(document.change_category.level2,"LGA-775",
"LGA-775");
AddOption(document.change_category.level2,"Socket-
AM2", "Socket-AM2");
}
...
function removeAllOptions(selectbox)
{
var i;
for(i=selectbox.options.length-1;i>=0;i--)
{
selectbox.remove(i);
}
}

function addOption(selectbox, value, text )
{
var optn = document.createElement("OPTION");
optn.text = text;
optn.value = value;
selectbox.options.add(optn);
}
</script>

```

A termék kiválasztása annak nevével lehetséges egy legördülő menüből, melynek tartalma az adatbázis jelenlegi állapotától függ. Törölt termék másik

kategóriába való áthelyezése nem lehetséges. A kategóriába való áthelyezést egy UPDATE művelettel végzem el, melyet a következő kódrészlet szemléltet:

```
$connection=new DBConnection();
$connection->query("SELECT * FROM product_list WHERE
category=".$id." AND name='".$$_POST["name"]."' AND
deleted='False' ORDER BY name ASC");
if (($numrows=$connection->getNumRows())==0)
{
    $connection->query("UPDATE product_list set
category=".$id." WHERE name LIKE
'".$$_POST["product"]."'");}
```

A kód ellenőrzi, hogy az adott kategóriában létezik-e már a kiválasztott nevű termék, és amennyiben nem, úgy frissíti a sort.

4. Kategória törlése

Bejelentkezve: [szolessz](#)
[Kijelentkezés](#)

- Új kategória
- Termék másik kategóriába sorolása
- Kategória törlése
- Új termék
- Termékek tulajdonságainak szerkesztése
- Új partner
- Partner szerkesztése
- Új gyártó
- Gyártó adatainak módosítása
- Új szállítvány
- Szállítványok megtekintése
- Adatbázis mentése
- Leadott rendelések megtekintése
- Boltos jogosultság
- Felhasználók leltársa, engedélyezése

Kategória törlése

Első szint: Második szint: Harmadik szint: Negyedik szint:

A kategória törlésénél is az előbbieken megismert működési elvű Javascript kód kerül felhasználásra. A törlés fizikai törlés jelent, tehát az adatbázisból véglegesen eltávolításra kerül az adott al- vagy főkategória. A törlésre való megszorítások a következők: nem lehet olyan kategóriát eltávolítani, melynek van alkategóriája, vagy valamilyen termék található az adott kategóriában. A törlés egy adatbázisbeli DELETE művelettel kerül megvalósításra.

5. Új termék hozzáadása

A menüpontban egy új termék hozzáadására kerül sor. Miután megadjuk a termék nevét, a kód ellenőrzi, hogy az adott kategóriában azonos néven szerepel-

e már termék. Amennyiben igen, azt jelzi és a termék bejegyzésére nem kerül sor. Akkor sem kerülhet sor új termék hozzáadására, ha az adott kategóriának van alkategóriája. Ebben az esetben is kapunk értesítést. A termék hozzáadása is INSERT művelettel történik.

Bejelentkezve: szelessz
Kijelentkezés

Új kategória
Termék másik kategóriába sorolása
Kategória törlése

Új termék
Termékek tulajdonságainak szerkesztése

Új partner
Partner szerkesztése

Új gyártó
Gyártó adatainak módosítása

Új szállítvány
Szállítványok megtekintése

Adatbázis mentése

Leadott rendelések megtekintése

Boltos jogosultság
Felhasználók letiltása, engedélyezése

Új termék hozzáadása

Első szint: Második szint: Harmadik szint: Negyedik szint:

Válassz Válassz Válassz Válassz

Termék neve:

OK

6. Termékek tulajdonságainak szerkesztése

A nyitókép, ami ebben a menüben fogad minket egy legördülő menü, melyből a termék nevét választhatjuk ki. Ezután egy *OK* nyomógombról való nyugtázást követően a fenti képhez hasonló láthatunk magunk előtt, melyben megadhatjuk az általunk kiválasztott termék nettó árát, áfa kulcsát, kiválaszthatjuk gyártóját az általunk előzőleg az adatbázisba felvitt gyártók közül, a termékre történő közvetlen hiperhivatkozást, a termék egy rövid ismertetését, illetve a termék fotóját is. Az oldal alján láthatjuk, hogy jelenleg milyen fénykép van a termékhez hozzárendelve. Minden egyes termékhez egy alapértelmezett kép van hozzárendelve, mely az *images* könyvtárban található *default_picture.png* néven.

Bejelentkezés: [szolessz](#)
 Kijelentkezés

Új kategória
 Termék másik kategóriába sorolása
 Kategória törlése

Új termék
 Termékek tulajdonságainak szerkesztése

Új partner
 Partner szerkesztése

Új gyártó
 Gyártó adatainak módosítása

Új szállítvány
 Szállítványok megtekintése

Adatbázis mentése

Leadott rendelések megtekintése

Boltos jogosultság
 Felhasználók letiltása, engedélyezése

Termék tulajdonságainak szerkesztése

Nettó ár:

Áfa:

Gyártó:

Hivatkozás:


Kép:

Törölve:

Termékleírás:

Ez egy szép alaplap.

Jelenlegi kép:



Amennyiben az adott termékről választunk fotót, úgy nekünk kell gondoskodni róla, hogy megfelelő méretű és tájolású képet válasszunk. Tartózkodjunk a 400x300-as befoglaló kerettől nagyobb méretű képektől. A kép lehet gif, jpeg illetve png formátumú, neve lényegtelen, ugyanis a kiválasztás után az *images* könyvtárban a termék nevével megegyező néven kerül eltárolásra az eredeti fájlal megegyező kiterjesztéssel.

A törölve választónégyzet bejelölése esetén a termék logikai törlésre kerül, melyet a *PRODUCT_LIST* tábla *DELETED* mezőjének a TRUE értékre állításával érek el. Ez azt eredményezi, hogy a felhasználók nem fogják látni az adott terméket, így azt nem fogják tudni megvásárolni. A termék tulajdonságainak frissítése UPDATE művelettel történik.

7. Új partner hozzáadása

A partner egy olyan kereskedelmi egységet jelent, akitől a termékeket szállítmányok formájában a bolt megvásárolja továbbértékesítés céljából.

Bejelentkezve: szelessz
Kijelentkezés

Új kategória
Termék másik kategóriába sorolása
Kategória törlése

Új termék
Termékek tulajdonságainak szerkesztése

Új partner
Partner szerkesztése

Új gyártó
Gyártó adatainak módosítása

Új szállítmány
Szállítmányok megtekintése

Adatbázis mentése

Leadott rendelések megtekintése

Boltos jogosultság
Felhasználók leltára, engedélyezése

Új partner hozzáadása

Név:

Település:

Utca:

Házzszám:

Irányítószám:

Telefonszám:

E-mail cím:

Adóazonosító szám:

Partner hozzáadása

A partner megfelelő adatainak kitöltését követően a partner az adatbázisba bejegyzésre kerül, így ezután már tudunk tőle vásárolni. Két azonos nevű partner nem lehet az adatbázisban. A hozzáadás INSERT művelettel történik.

8. Partner szerkesztése

A képernyőn egy lista fogad minket az aktuális partnerek nevével. Amennyiben a megfelelő nevű partnerre kattintunk, úgy az *Új partner hozzáadása* menüpontból megismert űrlap fogad minket a partner aktuális adataival kitöltve, melyeknek módosításával aktualizálhatjuk az adatbázist egy UPDATE művelet segítségével.

9. Új gyártó hozzáadása

Bejelentkezve: [szelessz](#)
[Kijelentkezés](#)

- Új kategória
 - [Termék másik kategóriába sorolása](#)
 - [Kategória törlése](#)
- Új termék
 - [Termékek tulajdonságainak szerkesztése](#)
- Új partner
 - [Partner szerkesztése](#)
- Új gyártó
 - [Gyártó adatainak módosítása](#)
- Új szállítmány
 - [Szállítmányok megtekintése](#)
- [Adatbázis mentése](#)
- [Leadott rendelések megtekintése](#)
- Boltos jogosultság
 - [Felhasználók letiltása, engedélyezése](#)

Új gyártó hozzáadása

Név:

Weboldal címe:

A gyártó felvitelére azért van szükség, mivel a terméknek van egy gyártója, amit ki kell választanunk, ha helyes termék adatait helyesen szeretnénk nyilvántartani.

Amennyiben a gyártó nem rendelkezik weboldallal, úgy az adatbázisban alapértelmezettként megadott *http://* érték kerül bejegyzésre.

10. Gyártó adatainak módosítása

A gyártó nevének a listából történő kiválasztása után a Gyártó hozzáadása menüpontban megismert űrlapot láthatjuk magunk előtt, mely az aktuális adatokkal van kitöltve. A folyamat során az adatbázisban UPDATE művelet történik.

11. Új szállítmány hozzáadása

Bejelentkezve: szelessz
Kijelentkezés

Új kategória
Termék másik kategóriába sorolása
Kategória törlése

Új termék
Termékek tulajdonságainak szerkesztése

Új partner
Partner szerkesztése

Új gyártó
Gyártó adatainak módosítása

Új szállítmány
Szállítmányok megtekintése

Adatbázis mentése

Leadott rendelések megtekintése

Boltos jogosultság
Felhasználók letiltása, engedélyezése

Új szállítmány hozzáadása

Szállítólevél száma:

Dátum:

Partner:

Termék neve:

Darabszám:

Ár:

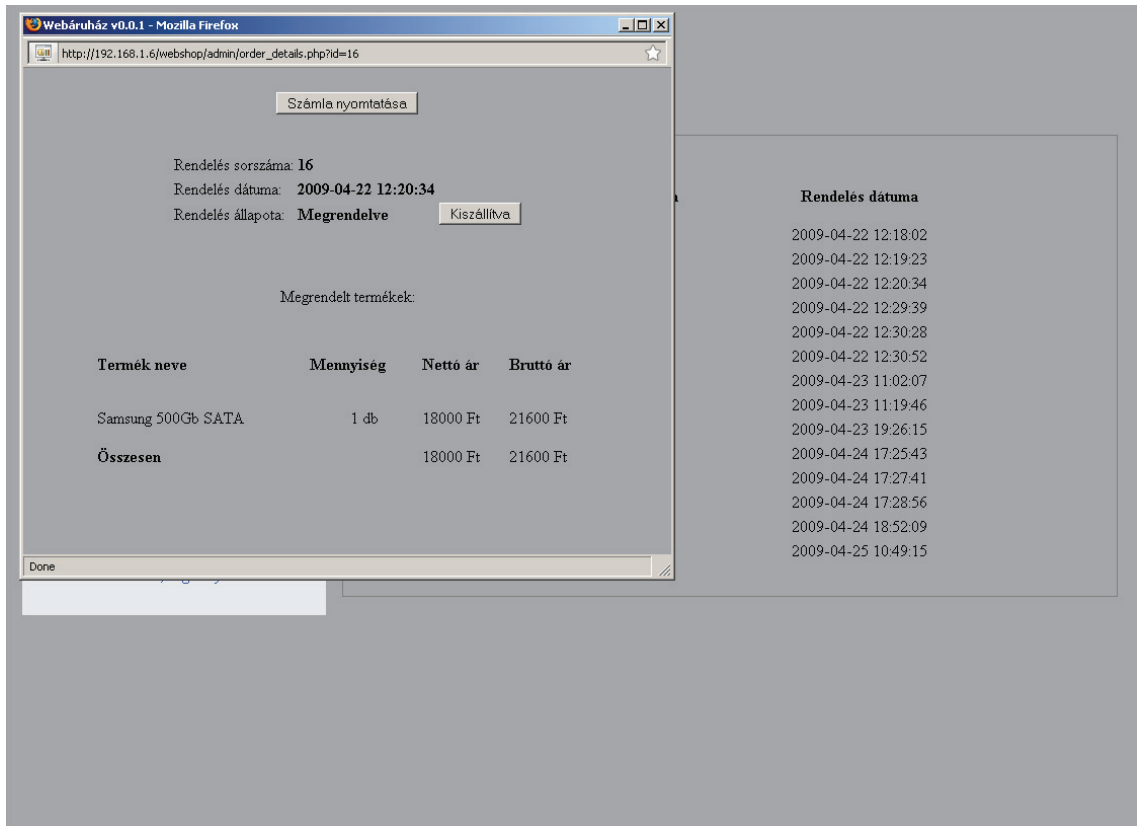
Megnevezés	Mennyiség	Ár
Western Digital 250GB SATA2	2	34000
Samsung 500Gb SATA	6	32500

A menübe lépve az alábbi kép fogad minket az oldal alján látható táblázatot kivéve. A szállítólevél száma egy karaktersorozat, melynek egyedinek kell lennie, ha már létező számot írunk be, akkor hibaüzenetet kapunk. A dátum automatikusan kerül kitöltésre, hiszen általában a szállítmány annak átvételének a napján kerül felvitelre. Természetesen a dátum módosítható. A partner nevének kiválasztása elegendő az utolsó szállítmánybeli termék hozzáadása után.

A termék nevének, darabszámának és annak árának megadása után a *Termék hozzáadása* gombra kattintással elérjük, hogy az alul lévő táblázat egy új sorral bővül.

Ha minden szállítmánybeli terméket felvittünk, úgy az *OK* gombra kattintva a szállítmány az adatbázisba kerül, illetve minden egyes termék egyedi azonosítót kap a *PRODUCTS* táblában. A dolog nehézségét az adja, hogy több táblában kell egymás után végrehajtani a különböző módosításokat.

12. Leadott rendelések megtekintése



Számola nyomtatása

Rendelés sorszáma: 16
Rendelés dátuma: 2009-04-22 12:20:34
Rendelés állapota: **Megrendelve**

Megrendelt termékek:

Termék neve	Mennyiség	Nettó ár	Bruttó ár
Samsung 500Gb SATA	1 db	18000 Ft	21600 Ft
Összesen		18000 Ft	21600 Ft

Done

Rendelés dátuma

- 2009-04-22 12:18:02
- 2009-04-22 12:19:23
- 2009-04-22 12:20:34
- 2009-04-22 12:29:39
- 2009-04-22 12:30:28
- 2009-04-22 12:30:52
- 2009-04-23 11:02:07
- 2009-04-23 11:19:46
- 2009-04-23 19:26:15
- 2009-04-24 17:25:43
- 2009-04-24 17:27:41
- 2009-04-24 17:28:56
- 2009-04-24 18:52:09
- 2009-04-25 10:49:15

A képernyő jobb oldalán egy táblázat fogad minket, melyben a leadott, de még ki nem szállított rendelések azonosítóját és annak dátumát láthatjuk. A kívánt azonosítóra kattintva egy új ablakban jelennek meg a rendelés részletei. A Kiszállítva gombra kattintva a rendelés állapotát változtathatjuk meg megrendeltről kiszállítottra. A *Számola nyomtatása* gomb egy pdf-et generál TCPDF segítségével, mely egy nyílt forráskódú PHP-ben írt alkalmazás. Több programot is kipróbáltam, melyeknek ugyanezt a feladatot kellett volna ellátni, de ez volt az egyetlen olyan ingyenes, mely képes volt UTF-8 kódolású szöveget feldolgozni. Bár beépített betűtípusainak lehetővé kellett volna tenni a magyar ABC összes ékezetes karakterének helyes megjelenítését, az é, ő, ű, í betűk helyett csak egy négyzet volt látható a generált pdf fájlban, ami viszont egyszerűen orvosolható volt a megfelelő betűtípus fájljainak (*times.ttf*, *timesbd.ttf*, *timesi.ttf*, *timesbi.ttf*) a TCPDF számára emészthető formába való alakításával. A pdf a böngésző beállításaitól függően vagy automatikusan

megnyitásra kerül, vagy a böngésző felkínálja, hogy mentse-e, vagy nyissa meg. A fájl ezektől függetlenül automatikusan a *szamlak* könyvtárban el lesz mentve. A pdf fájl első oldalán a számla szerepel, a második oldalon pedig a garancialevél. A számla tartalmazza a szükséges adatokon kívül áfa kulcsenkénti bontásban az áfa mértékét, az adóalap, az adó, illetve a bruttó összeg értékét is. A Függelékben megtekinthető a számla egy mintapéldánya. A számla kinézete az Ipon Kft számlalevelén alapul.

Amennyiben a számlát már létrehoztuk, a későbbiekben annak újbóli létrehozására már nincs lehetőség, ám annak megtekintésére igen.

13. Boltos jogosultság

A webáruházba külön boltos jogkörrel rendelkező felhasználó regisztrálására nincs lehetőség, ám a regisztrált felhasználó jogkörét a későbbiekben bővíthetjük, vagy csökkenthetjük.

A webáruházba alapértelmezettként felvitt *admin* nevű felhasználó jogkörét biztonsági okokból nem lehet megváltoztatni, hiszen egy véletlen félrekattintás folytán így teljesen kizárhatjuk magunkat a webáruház boltosoknak fenntartott részéből.

A megfelelő felhasználó név kiválasztását követően *Engedélyez* vagy *Tilt* feliratú gomb kerül megjelenítésre az aktuális jogkörtől függően és ezek valamelyikére kattintva érhető el a kívánt változtatás.

Miután a felhasználó boltos jogkörrel is rendelkezik, onnantól már a webáruház ebben a dolgozatnak ebben a szakaszában ismertetett részébe is beléphet. Ez a *USERS* tábla megfelelő mezőjének a módosítását jelenti.

14. Felhasználók letiltása, engedélyezése

Ebben a menüpontban a már regisztrált felhasználók számára tudjuk engedélyezni, illetve tiltani az áruházba való belépést. Sajnos a felhasználókat nem mindig a jó szándék vezérli, így ha bármilyen visszaélést tapasztalunk, úgy

a felhasználót kitilthatjuk áruházunkból. Ez vonatkozik mind a vásárló, mind a boltos jogkörrel rendelkezőkre. A webáruház *admin* nevű felhasználóját nem tilthatjuk le az előző pontban ismertetett okok miatt.

15. Felhasználói adatok megváltoztatása és kijelentkezés

A képernyő bal felső sarkában található felhasználói névre kattintva egy űrlapot látunk magunk előtt, mely az aktuális adatokkal van kitöltve, s melynek módosításával aktualizálhatjuk adatainkat.

A Kijelentkezésre kattintva a kód megszüntet minden munkamenet azonosítót, majd visszairányít minket a bejelentkező képernyőre.

```
session_start();  
session_unset();  
session_destroy();  
header ("Location: index.php");
```

4.3 Eltávolítás

Gondoskodjunk róla, hogy a weboldal ne legyen elérhető a vásárlók számára. Ezután töröljük az összes kosár tartalmát, majd egy biztonsági mentés után az adatbázis törölhető.

A mentésnek köszönhetően a későbbiekben egy másik környezetben akár újra felépíthető a rendszer és onnan folytathatja a működését, ahol abbahagyta. Egyéb okokból is szükség lehet az adatbázisra a későbbiekben, pl.: ellenőrizhetjük, hogy mekkora árukészlet maradt, vagy az adóhatóság számára tudunk adatokkal szolgálni.

5. A webáruház felépítése a vásárlók szemszögéből

1. A nyitóképernyő

WEBSHOP Online áruház

Üdvözzük Önöket webáruházunkban!

Név:
Jelszó:
OK

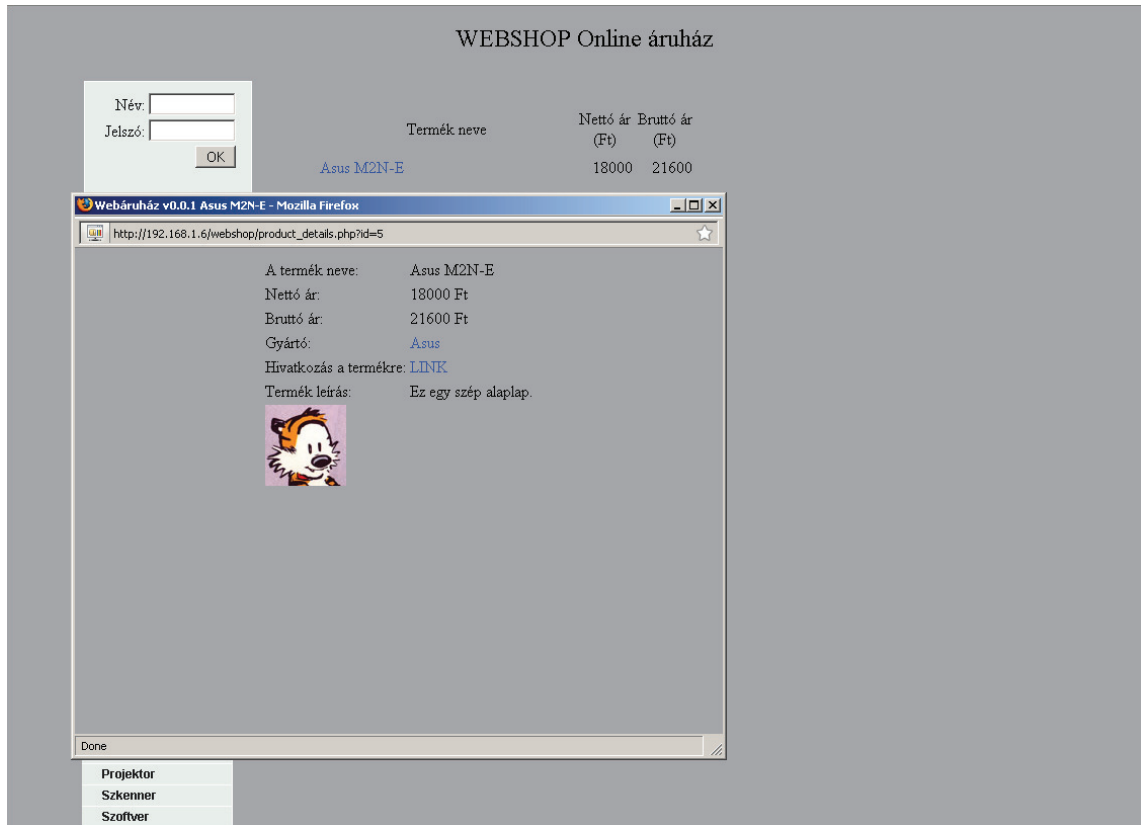
[Regisztráció](#)
[Elfelejtett jelszó](#)

- Alaplap
- Asztali számítógép
- DVD író
- Egér
- Egér+Billentyűzet
- Floppy
- Főkategória
- Játék
- Kábel
- Memória
- Memória kártya
- Merevlemez
- Mobile Rack
- Monitor
- Multimédia
- Notebook
- Notebook táska
- Nyomtató
- Patron
- PDA tartozékok
- Pendrive
- Processzor
- Projektor
- Szkennel
- Szoftver

A nyitóképernyő bal felső részén jelentkezhetünk be felhasználó nevünkkel és jelszavunkkal. Amennyiben még nem regisztráltunk a rendszerbe, úgy a *Regisztráció* hivatkozásra kattintva megtehetjük azt. Egy új képernyő fogad minket egy üres űrlapmezővel, melynek a *Cég neve* mezőn kívül az összes többi kitöltése kötelező. A regisztráció során a megadott email címre egy megerősítő levelet kapunk, melyben lévő linkre kattintva immár vásárolhatunk is az áruházból.

Amennyiben elfelejtettük korábbi jelszavunkat, úgy az *Elfelejtett jelszó* hivatkozásra kattintva megadhatjuk email címünket, melyet a regisztráció során megadtunk, melyre egy új, a rendszer által véletlenszerűen generált jelszót kapunk. A jelszót érdemes a belépés után azonnal egy általunk észben tarthatóra változtatni.

```
$hash="";
$chars="abcdefghijklmnopqrstuvwxyz0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";
for($i=0;$i<16;$i++)
    $hash.=$chars[rand(0,strlen($chars))];
```



Regisztrálatlanul az áruházban található termékek megtekintésére van csak lehetőségünk vásárlás nélkül. A bal oldali többszintű menü egy lista, mely CSS segítségével érte el jelenlegi formáját.

A kívánt termék kategóriára kattintva megtekinthetjük az abban a kategóriában szereplő termékeket, s azoknak minden tulajdonságát.

A termék tulajdonságain belül a gyártó weboldalára, illetve a termék közvetlen oldalára juthatunk.

2. Bejelentkezés után

A bejelentkezett felhasználók számára megjelenik a jobb oldalon a kosár, illetve az előző rendelések menü. A termék kategóriát kiválasztva pedig feltárul a lehetőség a kosárba helyezésre is. Miután kiválasztottuk a megfelelő termékeket, majd mindből a megfelelő darabszámút a *Kosárba tesz* gomba kattintva a kosárba helyeztük, a *Kosár tartalma* gombra kattintva megtekinthetjük kosarunk aktuális tartalmát.

The screenshot displays the WEBSHOP Online áruház interface. On the left, there is a navigation menu with categories such as Alaplap, Asztali számítógép, DVD író, Egér, Egér+Billentyűzet, Floppy, Főkategória, Játék, Kábel, Memória, Memória kártya, Merevlemez, Mobile Rack, Monitor, Multimédia, Notebook, Notebook táska, Nyomtató, Patron, PDA tartozékok, Pendrive, Processzor, Projektor, Szkenner, and Szoftver. The main content area features a table with the following columns: Termék neve, Nettó ár (Ft), Bruttó ár (Ft), and Mennyiség. Two items are listed: Samsung 500Gb SATA (Nettó ár: 18000, Bruttó ár: 21600, Mennyiség: 1 db) and Western Digital 250GB SATA2 (Nettó ár: 15000, Bruttó ár: 18000, Mennyiség: 1 db). Each item has a 'Kosárba tesz' button. On the right side, there is a 'Kosár tartalma' section showing 'A kosárban 4 db termék van' and an 'Előző rendeléseim' button. A top-left box contains the text 'Üdvözlöm, szelepsz' and 'Kijelentkezés'.

Termék neve	Nettó ár (Ft)	Bruttó ár (Ft)	Mennyiség
Samsung 500Gb SATA	18000	21600	1 db
Western Digital 250GB SATA2	15000	18000	1 db

Itt két lehetőség közül választhatunk. Vagy ürítjük a kosarunkat, vagy a *Megrendel* gombra kattintva egy újabb oldalra jutunk, ahol a *Megrendel* gombbal véglegesíthetjük rendelésünket. A rendelés sorszámáról értesítést kapunk mind a képernyőn, mind emailben küldött üzenet formájában az általunk regisztrált címen. A rendelés leadását követően a kosár tartalma automatikusan kiürül. Amennyiben valamilyen hiba folytán a kosárba helyezést követően a weboldallal

megszakad a kapcsolat, vagy kijelentkezünk, úgy az újbóli bejelentkezéskor kosarunk a legutóbbi tartalmával fogad minket.

Rendeléseink állapotáról az *Előző rendeléseim* gombra kattintva tájékozódhatunk.

The screenshot shows a web shop interface titled "WEBSHOP Online áruház". On the left, there is a navigation menu with "Üdvözlöm, szeleszz" and "Kijelentkezés". In the center, a list of orders is shown under the heading "Rendelés dátuma":

	Rendelés dátuma
1.	2009-04-22 12:18:02
2.	2009-04-22 12:19:23
3.	2009-04-22 12:20:34
4.	2009-04-22 12:29:39

On the right, there are two buttons: "Kosár tartalma" (A kosár üres) and "Előző rendeléseim". The "Előző rendeléseim" button is highlighted, and a detailed view of an order is shown in a browser window. The browser window title is "Webáruház v0.0.1 - Mozilla Firefox" and the URL is "http://192.168.1.6/webshop/previous_order_details.php?id=31". The order details are:

Rendelés sorszáma: 31
Rendelés dátuma: 2009-04-27 09:14:20
Rendelés állapota: **Megrendelve**

Megrendelt termékek:

Termék neve	Mennyiség	Nettó ár	Bruttó ár
Western Digital 250GB SATA2	3 db	15000 Ft	18000 Ft
Samsung 500Gb SATA	1 db	18000 Ft	21600 Ft

At the bottom of the browser window, the status "Done" and "Szoftver" are visible.

Az összes előzőleg leadott rendelésünk részletét megtekinthetjük annak dátumára való kattintással.

6. Összegzés

A webáruház fejlesztése során a kitűzött célokat sikerült megvalósítani, közelebbről megismerkedtem a PHP és PostgreSQL lehetőségeivel, funkcióival, korlátaival, melyeket későbbi munkám során alkalmazhatok.

A rendszer nem tökéletes, kisebb hibák biztosan előfordulnak benne, ám az adatbázis interfésznek köszönhetően akár másik programnyelvre, akár másik adatbázisrendszerbe viszonylag könnyen implementálható és a moduláris felépítésnek köszönhetően új funkciókkal egyszerűen bővíthető.

7. Felhasznált irodalom

Moulding, Peter: PHP Haladóknak Fekete Könyv, Perfact Pro Kft, 2002.

PHP Documentation Group – PHP kézikönyv, www.php.net, 1997-2009.

PostgreSQL 8.3.7 Documentation, www.postgresql.org, 2009.

Apache Software Foundation – Apache Manual, www.apache.org, 2002.

TCPDF Documentation, www.tcpdf.org, 2002-2009.