

# Tartalomjegyzék

1. Bevezetés .....	2
2. Online kereskedelem .....	3
3. Feladat specifikáció, elemzés .....	7
3.1. Webáruház .....	7
3.2. Az alkalmazással szembeni elvárások .....	8
4. Tervezés, részfeladatokra bontás.....	9
4.1. Adatbázis megtervezése .....	9
5. Felhasználói felület megtervezése.....	16
6. A webáruház általános felépítése .....	18
6.1. Vevői oldal.....	18
6.2. Eladói oldal .....	18
7. Megvalósítás (kódolás).....	19
7.1. Vevői oldal.....	19
7.1.1. Belépés.....	19
7.1.2 Regisztráció .....	20
7.1.3. Vásárlás .....	21
7.2. Eladói oldal .....	22
7.2.1. Belépés.....	22
7.2.2. Regisztráció .....	22
7.2.3. Kategória szerkesztése .....	22
7.2.4. Áru szerkesztése .....	22
7.2.5. Megrendelések.....	23
7.2.6. Számlázás .....	24
8. Üzembe helyezés, tesztelés .....	24
9. Összefoglalás .....	26
10. Felhasznált irodalom .....	28

## 1. Bevezetés

Napjainkban az internetes kereskedelem egyre nagyobb méreteket ölt. Az informatikával és más tevékenységgel foglalkozó cégeknek fontos, hogy interneten is el tudják érni őket. Ha egy cég nincs jelen az Interneten, jelentős méretű vásárlói tömegtől esik el.

Fontos, hogy az oldal megjelenésében, kialakításában elnyerje az ügyfelek tetszését. A jól megtervezett oldal, mely látványos de mégsem giccses, érthető, könnyen kezelhető, átlátható, emellett naprakész információkat és kiemelkedően jó minőségű szolgáltatásokat is tartalmaz. Sokkal könnyebben csábítja vissza a vásárlókat, mint bármelyik reklám.

Számos internetes bevásárlóközpont jelent meg az elmúlt években, de mivel a kapcsolatot az ügyfelekkel interneten és a posta felhasználásával tartották, ha garanciális probléma merült fel, még ha a közelben is volt a webáruház székhelye, akkor is ki kellett várni a szállítási és egyéb időszakot.

Szakedolgozatom alapja egy olyan adatbázis elérésű, dinamikus oldal létrehozása, mely az internetes vásárlást szolgálja. Nem csak a virtuális világban, hanem valójában is létezik egy bolt, ahol pillanatok alatt ki lehet szolgálni az ügyfeleket.

A PHP programozási nyelvet találtam a feladat megoldására a legalkalmasabbnak. Ez az egyik legnépszerűbb kiszolgáló oldali nyelv. Apache webservert és a MYSQL adatbázisszervert használtam fel.

Azért választottam ezt a témát, mert úgy érzem az internetben van a jövő és az online kereskedelemben. Ennek az oldalnak a megtervezésével lehetőségem nyílik tudásom tesztelésére és egy újabb szeletet megismerni az informatika világából.

## 2. Online kereskedelem [8]

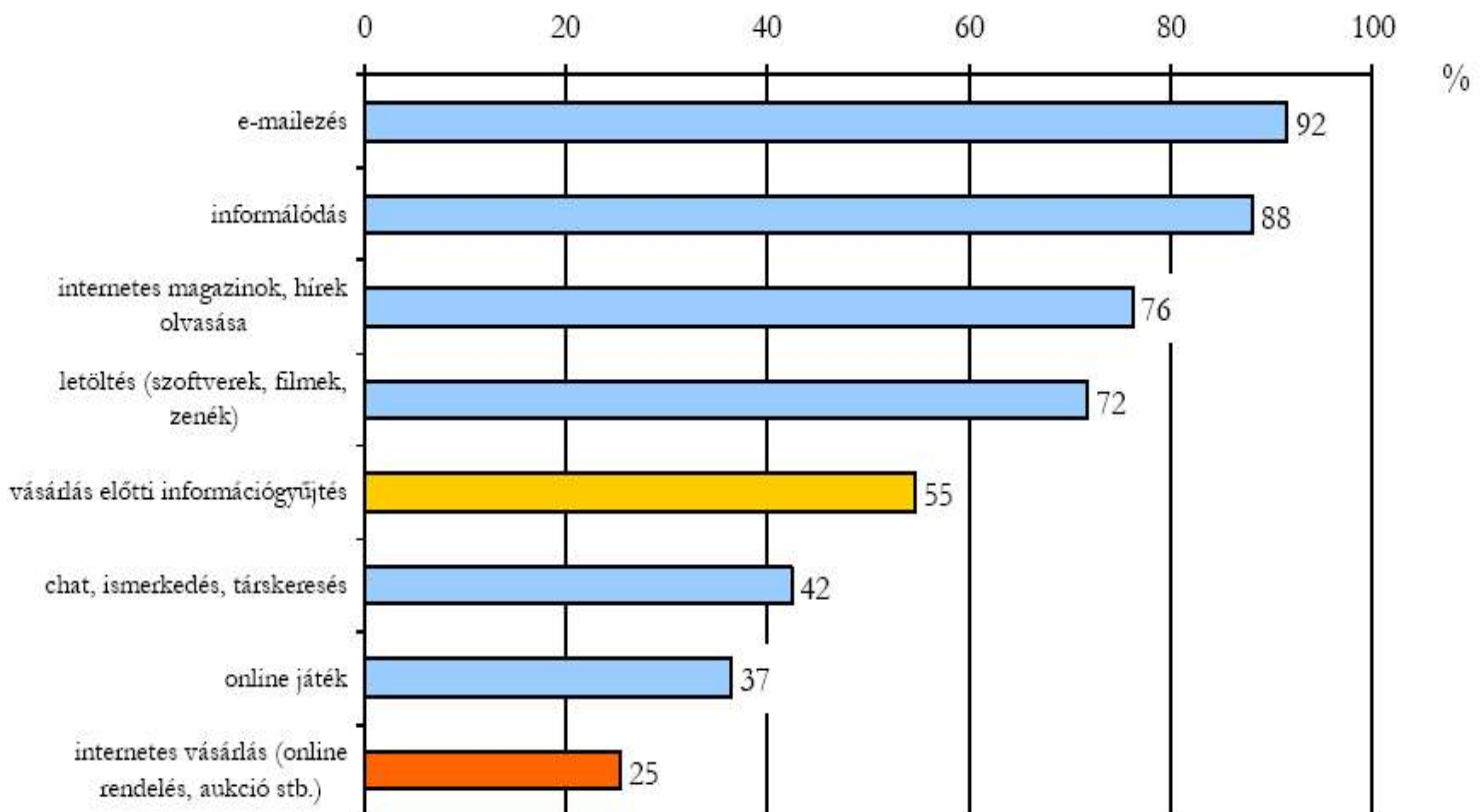
Az elektronikus kereskedelem alig néhány éves múltra tekinthet vissza. Egyelőre nincsen kialakult és elfogadott fogalomrendszere még fő közegében, az angol nyelvben sem. Az elektronikus csatornákat felhasználó, számítógép és telekommunikációs technológiára épülő üzleti tevékenységet **elektronikus kereskedelemnek**, angol szakkifejezéssel e-commerce-nek (electronic-commerce) hívjuk.

Hatalmas mennyiségű üzleti tranzakciót és információcserét lehet automatizálni ily módon a földrajzi akadályok leküzdésével. Az online vásárlás és eladás kifejezések arra a folyamatra utalnak, amikor a termékek, illetve szolgáltatások vétele illetve eladása kifejezetten az Interneten keresztül történik. Az IT – és az e-kereskedelem alkalmazásai – az üzleti/gazdasági tevékenységek alapvető elemeivé válnak és mélyreható változásokat generálnak a szervezetek struktúrájában, működésében és vezetésében is.

Felfogásunk szerint az e-kereskedelem fogalma a termékek, szolgáltatások, információk elektronikus hálózatokon keresztül történő cseréjének, értékesítésének folyamatát jelenti. Több perspektíva szerint is definiálható:

- Kommunikációs perspektíva: információ, termékek, szolgáltatások nyújtása vagy a fizetési folyamat bonyolítása valamilyen elektronikus csatornán keresztül.
- Üzleti folyamat perspektíva: a technológia alkalmazása az üzleti folyamatok és tranzakciók automatizálására.
- Szolgáltatási perspektíva: a szolgáltatások költségének csökkentését, minőségjavítás és a gyorsaság növelése mellett elérni célzó törekvések eszköze.
- Online perspektíva: képesség a termékek/szolgáltatások eladására és megvásárlására az Interneten vagy más on-line szolgáltatásokon keresztül.

## Mire használja az internetet?

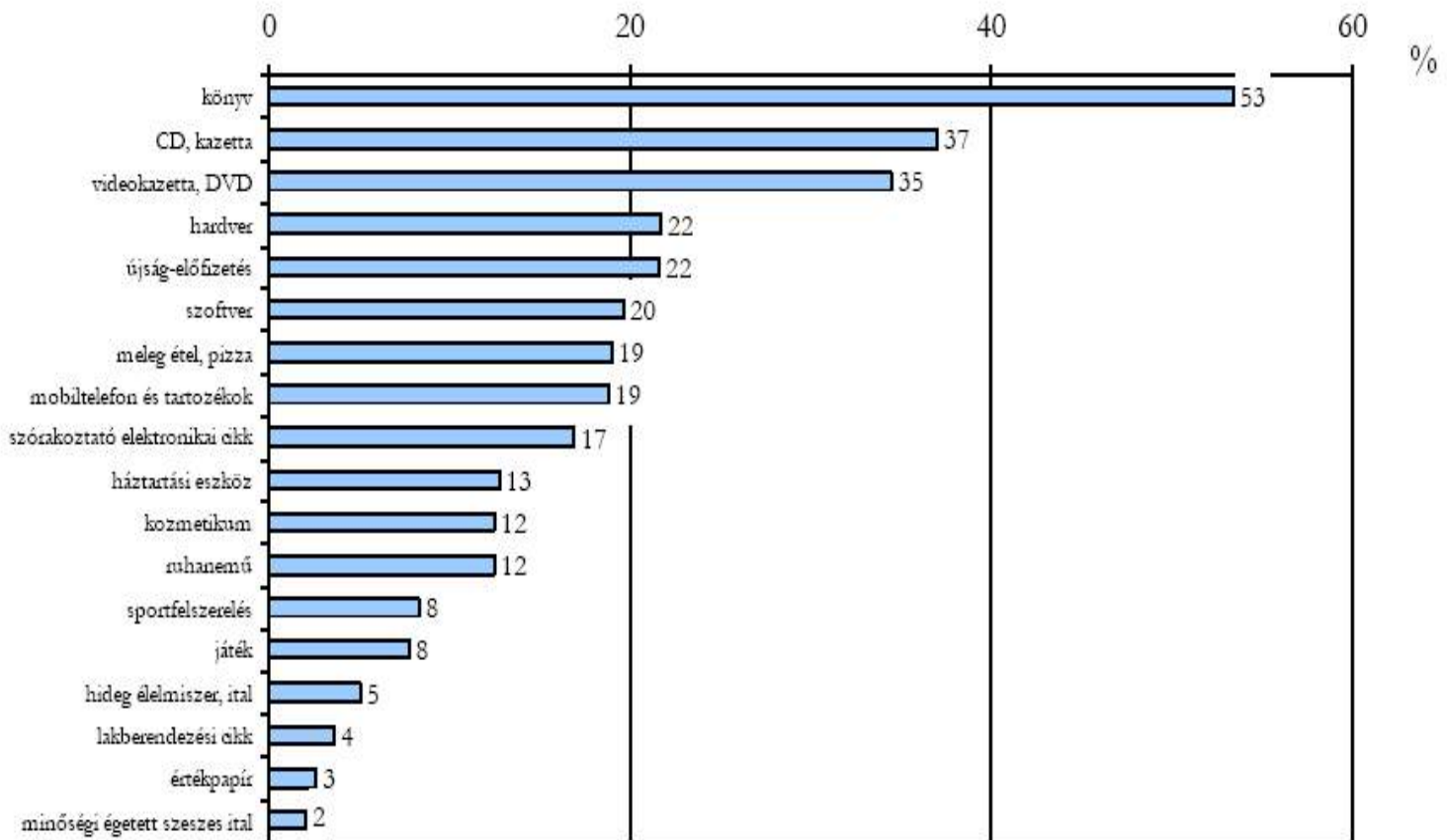


1. ábra, az internet felhasználása

Forrás: NCR Piackutató, 2004.

## Milyen termékek vásárlását tervezi az interneten?

*az online vásárlást biztosra ígérők körében*



2. ábra, internetes vásárlások

Forrás: NCR Piackutató, 2004.

Az online kereskedelem egyre nagyobb méreteket ölt. Sajnos Magyarországon az emberek még nem bíznak meg az internetben, ezért ez a fejlődés itt nem olyan gyors.

Számos érv szól az online kereskedelem mellett és természetesen ellene is. Egy online webáruházat viszonylag kis befektetéssel létre lehet hozni, nem igényel nagy fizikai területet.

A költségek is kisebbek, mert:

- nem kell készletezni
- magas szintű automatizmus
- önkiszolgálás
- kis munkaerőigény, hatékony erőforrás gazdálkodás

A szakdolgozati témám szerint a vállalkozás kezdeti szakaszában elég egy ember, aki ellátja az offline (valós) és online kereskedelmet. Ha növekszik az internetes megrendelések száma, akkor már legalább két alkalmazottra lehet kiosztani az egyes feladatokat.

Egy webáruháznál a gyorsabb információáramlás miatt, a legújabb termékekhez előbb hozzájuthatnak a vevők. Mondhatni, hogy nincsenek korlátok a virtuális piacon.

Különbség online és offline kereskedelem között:

Vevőkezelés

- Online: egy regisztrációval mindent tudunk a vevőről
- Offline: bejön, választ, fizet és elmegy az áruval

Kínálat

- Az van, ami a polcra ki van téve
- Minden megrendelhető, ami az adatbázisban van, csak fel kell tölteni

0-24 óráig nyitva tartó üzletekben a választék bővülésével a vásárlás, a szolgáltatás elérése sokkal kényelmesebb. A virtuális boltban gyorsan és kényelmesen választhatja ki fogyasztói igényeinek megfelelően a termékeket.

### 3. Feladat specifikáció, elemzés

A munkafolyamat megtervezése fontos. Egy jól megtervezett munkavégzés nagymértékben felgyorsíthatja a kivitelezést. A részfeladatok funkciójuk szerint kerülnek megalkotásra. Így könnyen lehet ellenőrizni a működésüket.

A portál megvalósítása, a megvalósítás részletlépésekre bontása a következő:

#### **A rendszer általános működésének vázlata**

A működési vázlat arra szolgál, hogy mielőtt még bármihez is hozzáfognánk, kialakuljon bennünk egy kép a végleges formáról. Ezek magukba foglalják a szerkezeti felépítést és a megjelenést.

#### **Adatbázis megtervezése teljes optimalizálással, a leghatékonyabb működés érdekében**

Az adatbázis az alapja a webáruháznak az összes adatot itt tároljuk. Megtervezésekor a legalaposabbnak kell lenni, mert később a változtatás nehezen eszközölhető.

#### **Felhasználói felület kialakítása**

A felhasználói felületnek kényelmi szempontból a lehető legegyszerűbbnek és felhasználó barátinak kell lennie. Az emberek az alapján ítélnek, amit látnak.

#### **Kódolás, megvalósítás**

Megtervezések és az előzetes elképzelés alapján, le lehet programozni az oldalt. Sokkal egyszerűbb úgy, hogy már meg van a kialakult kép.

#### **Üzembe helyezés, tesztelés**

A tesztelés a végső fázisa is a kivitelezésnek, de a leghasznosabb fejlesztés közben tesztelni, mert akkor a hibákat azonnal ki lehet szűrni.

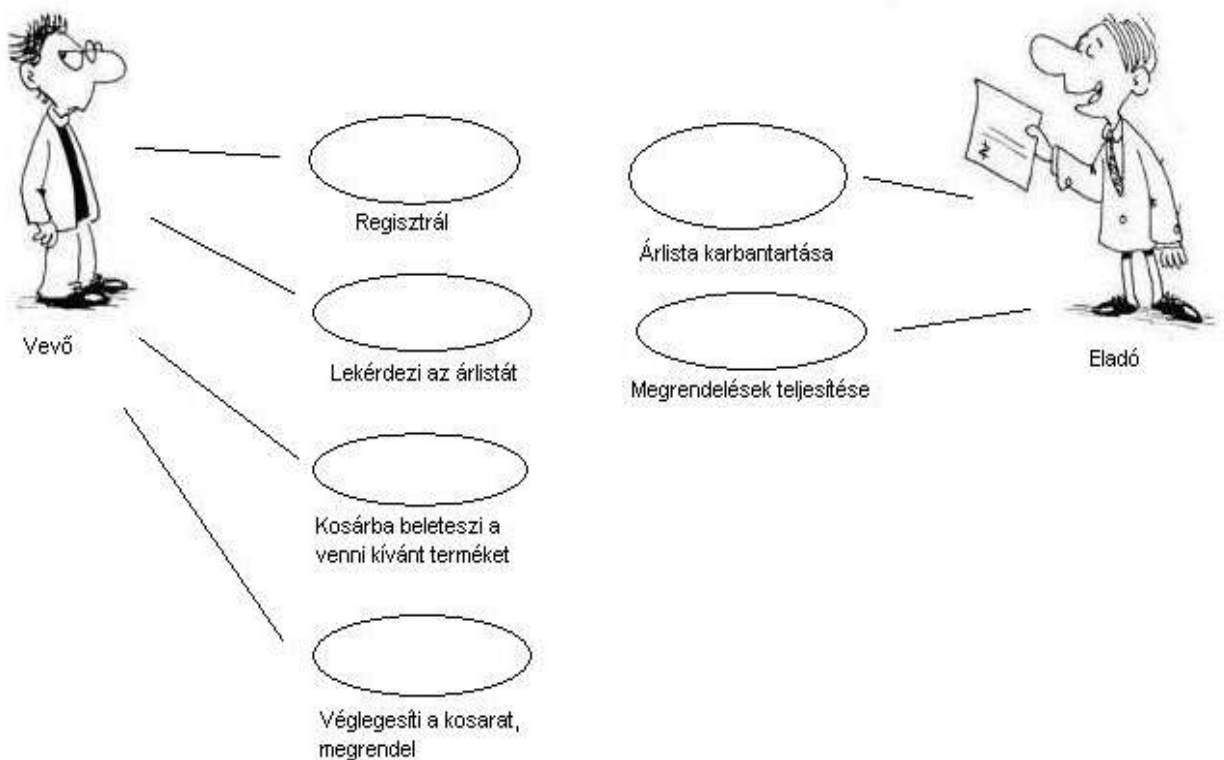
### 3.1. Webáruház

Az interneten történő böngészéseim során gyakran tapasztaltam, hogy az egyes webáruházakban szinte eltéved az ember, mert reklámokkal, különféle cikkekkel zsúfolt az oldal. Ezért igyekeztem a lehető legátláthatóbb és legegyszerűbb felhasználói felületet kialakítani.

Ez a webáruház csak informatikai eszközökkel foglalkozik. De a termékek típusok listája bővíthető. Bármely termékkel előfordulhat, hogy meghibásodik, és sokszor nagyon problémás a garancia elintézése, (sokan ezért se vásárolnak interneten) ezért úgy döntöttem, hogy online áruházhoz tartozzon egy offline bolt is, így a felmerülő problémákat könnyebben lehet kezelni. A két rész együtt, de mégis kicsit függetlenül működik egymástól.

### 3.2. Az alkalmazással szembeni elvárások

Az egész üzlet adatait egy MySQL adatbázisban tároljuk, melyet az internetről tud elérni egyaránt a vevő és az eladó is csak különböző felhasználói felületeken. Ennek a web-alapú rendszernek az előnye, hogy az eladó is, úgy mint a vevő bárholnan el tudja érni az adatbázist. . Emellett a mobilitás nagyon fontos.



3 ábra, Webáruház szerkezete



## 4. Tervezés, részfeladatokra bontás

### 4.1. Adatbázis megtervezése

Az adatbázis-tervezés egy folyamat, mely több lépésből tevődik össze. Először az adatbázisban leképezendő rendszert elemzésnek vetjük alá. Meghatározzuk a tárolandó adatok körét, azok egymásközi kapcsolatait és az adatbázissal szemben felmerülő igényeket. Ezután következik a rendszertervezés, melynek eredménye az adatbázis logikai modellje.

Végül fizikai szinten képezzük le a logikai adatbázis modellt a felhasználható szoftver és hardver függvényében. Ahhoz, hogy egy jól szervezett, jól működő adatbázist kapjunk, amiből mindig pontosan azt az adatot kapjuk vissza, amelyiket szeretnénk, akkor a tervezésnél nem csak az adatokat kell táblákba rakosgatni, hanem nagyon komolyan kell figyelni néhány egyszerűnek tűnő dolagra.

Az adatbázis tartalmazza online felhasználók, eladók, termékek, megrendelések, számlák adatait. A számlák adatai külön vannak választva az internetes megrendelésektől, későbbi könyvelési és átláthatósági szempontból lényeges.

Minden táblának külön azonosítója van, ezekkel hivatkozunk egymásra. A felhasználókat, vagyis az online megrendelőket a „felhasznalo” táblába tároljuk. Ezen adatok alapján állíthatjuk ki a megrendelésekről a számlát és végezhetjük az anyagi jellegű tranzakciókat.

Az eladók részére is található egy tábla, melyben csak pár fontos adatot rögzítünk. Ezt a táblát fel is töltjük a főnök adataival, mert a rendszer úgy működik, hogy csak egy regisztrált dolgozó tud egy újabb munkatérrel felvenni.

Az áruhoz tartozó rendszer két részből áll, az egyik a „kategoria” a másik az „aru” tábla. Ezek kapcsolatban vannak egymással, hisz a felhasználó először a kategóriát választja ki és utána a hozzátartozó árut.

Egy átmeneti adatok tárolására létrehozott tábla a kosár. Ebben csak addig vannak a rendelésre szánt termékek, míg a vevő nem véglegesíti a kosár tartalmát.

A „kosar” és a „megrendelesek” tábla szoros kapcsolatban van egymással. A kosár tartalma ugyanúgy belekerül ebbe a táblába is. A véglegesítéskor ennek tartalma nem

törlődik, hanem kap egy állapotot „folyamatban”névvel és ez akkor változik át „feldolgozva”-ra, ha az eladó teljesíti a megrendelést.

A fent említett táblák az online vevőkkel foglalkoznak. Az offline vásárlók részére két tábla kell. Az egyik az offline vevőket tartja számon, a másik az egyes vásárlók megvett tételeit.

A fizikai tervezés során leginkább arra kell koncentrálni, hogy a logikai szerkezet mennyire felel meg a hatékony végrehajtás feltételeinek, illetve milyen indexeket rendeljünk az egyes relációkhoz. A relációkon végrehajtott művelet együttest tranzakciónak nevezzük. Általában a tranzakciók gyors végrehajtását kívánjuk elérni.

**Adatbázis szerkezet:**

Tábla neve:	felhasznalo
Felhasználó azonosító	<b>azon</b> int (11)
Név	felh_nev varchar (50)
Login	felh_login varchar (10)
Kód	felh_kod varchar (75)
Születési dátum	felh_szul date
Személyi igazolvány szám	felh_szemigysz varchar (15)
Ország	felh_oroszag varchar (50)
Irányítószám	felh_irszam int (4)
Város	felh_varos varchar (50)
Út	felh_ut varchar (50)
Hákszám	felh_hazsz int (3)
Bankszámlaszám	felh_banksz. varchar (50)
Telefonszám	felh_tel. varchar (12)
Vásárlást ellenőrző állapot	vette enum('nincs','van')

Tábla neve:	kategoria
Kategória azonosító	<b>azon</b> int(11)
Név	kat_nev varchar (50)
Leírás	kat_leir text

Tábla neve:	aru
Áru azonosító	<b>azon</b> int (11)
Hivatkozás a kategória azonosítóra	kat_azon int (11)
Név	aru_nev varchar (50)
Ár	aru_nettoar float (8,2)
Mennyiség	aru_menny int (4)
Kép	aru_kep varchar (75)
Leírás	aru_leir text

Tábla neve:	kosar
Kosár azonosító	<b>azon</b> int (11)
Hivatkozás a felhasználó azonosítóra	felh_azon int (11)
Hivatkozás az áru azonosítóra	aru_azon int (11)
Megrendelt mennyiség	kos_menny int (4)
Kosárba helyezés dátuma	kos_datum date

Tábla neve:

Megrendelések azonosító  
 Hivatkozás a felhasználó azonosítóra  
 Hivatkozás az áru azonosítóra  
 Megrendelt áruk mennyisége  
 Megrendelés dátuma  
 Felhasználói feldolgozás folyamata  
 Eladói feldolgozás folyamata

megrendelesek
<b>azon</b> int (11)
felh_azon int (11)
aru_azon int (11)
kos_menny int(4)
kos_datum datetime
allapot enum ('folyamatban','feldolgozva')
allapotmegr enum ('folyamatban','feldolgozva')

Tábla neve:

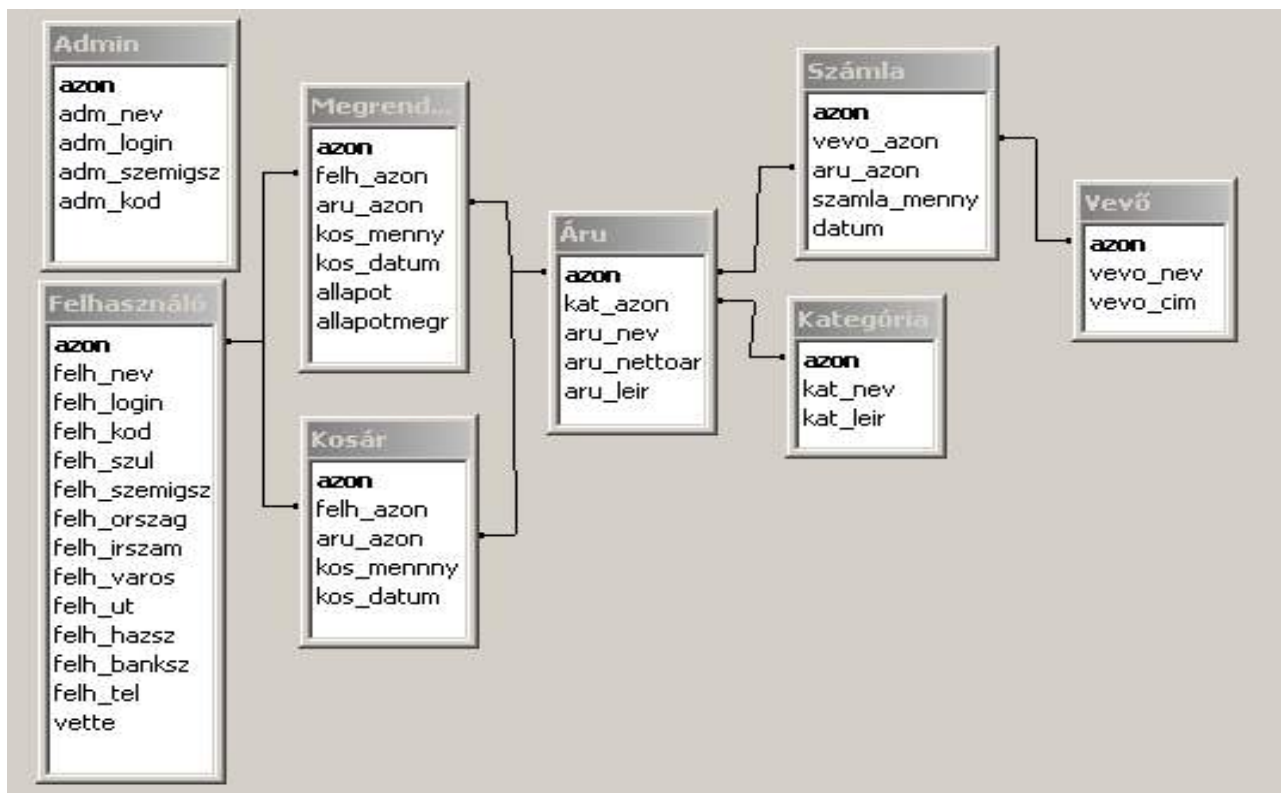
Vevő azonosító  
 Vevő neve  
 Vevő címe

vevo
<b>azon</b> int (11)
vevo_nev varchar (50)
vevo_cim varchar (50)

Tábla neve:

Számla azonosító  
 Hivatkozás a vevő azonosítóra  
 Hivatkozás az áru azonosítóra  
 Vett mennyiség  
 Dátum

szamla
azon int(11)
vevo_azon int (11)
aru_azon int (11)
szamla_menny int (11)
datum datetime



4. ábra, kapcsolatok

Az adatbázis megvalósításához szükséges a MySQL jelenléte. Az adatbázis a PHPMyAdminnal kényelmes grafikus felületével kerül felvitelre.

A MySQL egy többszálú, több felhasználós, erős SQL adatbázis szerver. A MySQL a kis és közepes méretű adatbázisoknál nagyon gyors. A fejlesztők erre különösen figyeltek. A MySQL Szervernek rengeteg olyan része van, amit a felhasználókkal közösen fejlesztettek ki, ezzel is növelve a használhatóságot. Mivel a MySQL nyílt forráskódú így a fejlesztés/fejlődés is jóval gyorsabb ütemben történik, mint a többi alkalmazás esetében. Tulajdonképpen a fejlődés folyamatos és szinte naponta jelenhetnek meg a frissebb - és valamilyen szempontból jobb – verziók.

A PHPMyAdmin napjaink egyik legnépszerűbb ingyenesen használható webes felülete, amely MySQL adatbázisokat kezel. A PHPMyAdmin tulajdonképpen nem más, mint egy PHP függvény gyűjtemény. Célja a MySQL adatbázisok Weben keresztül történő, vizuális kezelése. A megfelelő jogosultságok mellett (lekérdezési, módosítási, törlési jogosultság), a grafikus felületnek köszönhetően sokkal könnyebben áttekinthető, mint a karakteres felületen üzemelő programok. Egyik nagy előnye az egyszerűen végrehajtható, adat import és export.

Az ingyenes tárhelyet biztosító szolgáltatók túlnyomó többsége ezt a felületet használja, hogy biztosítsa az adatbázis-elérést a felhasználók számára. Ilyen szolgáltatók például az Ultraweb, SRV... stb. De ne higgyük, hogy csak az ilyen jellegű szolgáltatók, cégek használják ezt a felületet, hiszen kevés olyan alkalmazás, illetve felhasználói felület van, ami hasonlóan pozitív tulajdonságokkal rendelkezik.

A másik módszer a már elavultnak tűnhet, de aki 8-10 évvel ezelőtt már használt számítógépet, az lehet, hogy inkább a karakteres felületet fogja használni az adatbázis megvalósításához.

A WinMySQLAdmin.exe egy ablakos formában jeleníti meg a MySQL tulajdonságait. A data alkönyvtárban található meg adatbázisunk. A táblák neve háromféle végződéssel jelenik meg, mint például:

```
felhasznalo.frm (leíró file)
felhasznalo.MYD (adat file)
felhasznalo.MYI (index file)
```

A mysql.exe elindításakor egy parancssoros adatbeviteli alapokon működő ablak jelenik meg. A MYSQL> prompt után írhatjuk be az SQL parancsokat. (Ezek nem érhetők el, hogy ha egy ingyenes portálra regisztráljuk magunkat, ott csak a PHPMyAdmin-nal dolgozhatunk)

A MySQL szinte minden platformon működik, kezelése egyszerű, logikus és lényegretörő.

Az adatbázist adatokkal nem kell feltölteni, mert minden a webes felületen történik. Egyedül a tulajdonos adatait kell tartalmaznia.

A táblák, az adatok felvételét SQL nyelv használatával tudjuk megtenni. Az SQL (Structured Query Language – Strukturált lekérdező nyelv) egy a relációs adatbázisok létrehozására és lekérdezésére szolgáló, 1986 óta szabványosított nyelv. Elsősorban a relációkban (táblákban) tárolt adatok visszanyerésére szolgál, de emellett támogatja az adatok beírását és csoportosítását is. Ez a nyelv ún. eljárásmentes nyelv, tehát nem az adatok elérési módját, hanem a kívánt információ tulajdonságait kell megadni. Az SQL nyelvezete a köznapi beszédhez és gondolkodáshoz közel áll. Nem kell azzal foglalkoznunk, hogy az adatbázisból milyen algoritmusok alapján, hogyan kereshetők ki az adatok.

Példa egy tábla létrehozására:

```
create table aru (
  azon int not null primary key auto_increment,
  kat_azon int not null,
  aru_nev varchar (50) unique,
  aru_nettoar float (8,2),
  aru_menny int (4),
  aru_kep varchar (50),
  aru_leir text
);
```

Kapcsolat PHP és MySQL között:

A PHP forráskódunkba kell beleírni az adatbázishoz kapcsolódás parancsait. Erre szolgál a „mysql\_connect()” használata. Amint szükségünk van a MySQL-re, meg kell teremtenünk az összeköttetést.

Az összeköttetést létrehozó utasítás alakja.

```
mysql_connect ("hoszt neve", "felhasználónév", "jelszó");
```

Példa programból:

```
$kapcs=mysql_connect (DBHOSTNAME,DBUSERNAME,DBPASSWORD) or  
die (mysql_error (hiba));
```

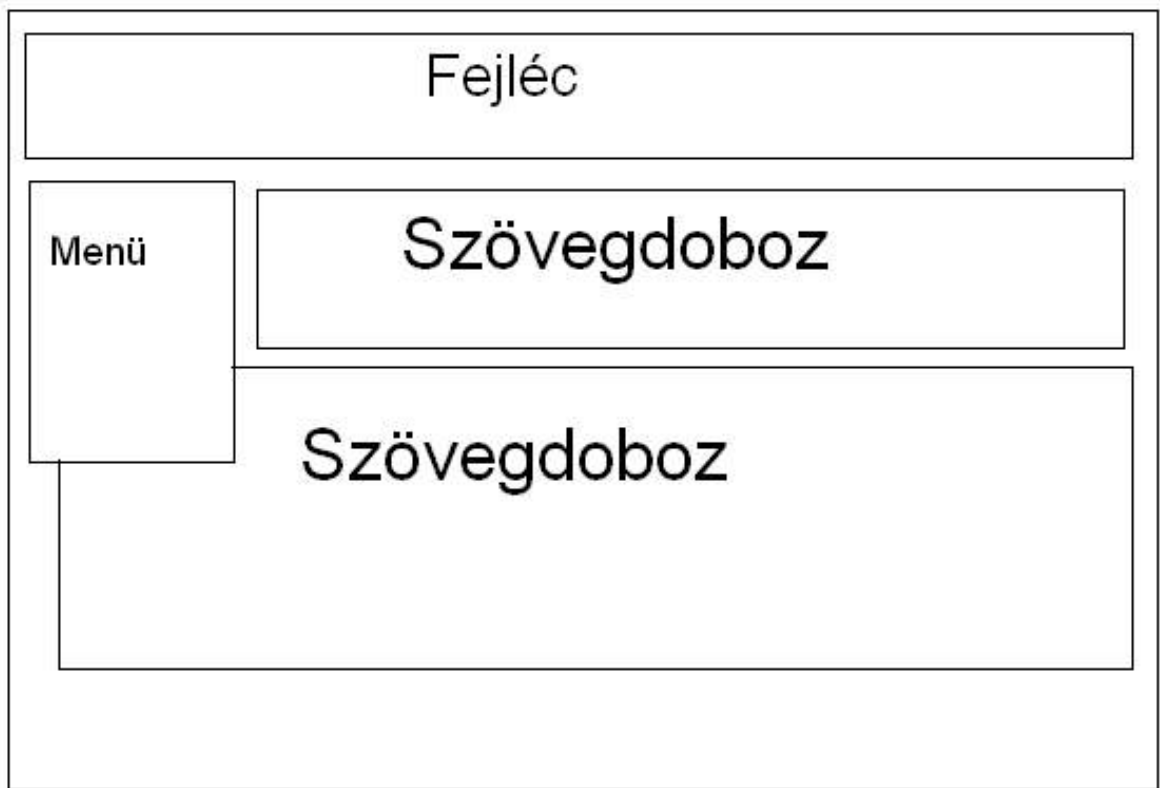
```
mysql_select_db (DBNAME,$kapcs) or die (mysql_error ());
```

Ebben a kódban kiválasztja az adatbázist is. Az egyszerűség kedvéért létrehoztam egy „db.php” állományt mely tartalmazza a kapcsolódás adatait. Ha megváltozik a környezet, elég csak ennek a file-nak a tartalmát átírni, ugyanis a többi állományban erre hivatkozunk:

```
include("db.php");
```

```
<?php  
define ("DBHOSTNAME", "host");  
define ("DBUSERNAME", "felhasználónév");  
define ("DBPASSWORD", "jelszó");  
define ("DBNAME", "adatbázis");  
?>
```

## 5. Felhasználói felület megtervezése



5. ábra, honlap szerkezete

A felhasználó felület megtervezésekor, a legegyszerűbb és a legátláthatóbb szerkezetet használtam fel. A grafikai megjelenítéshez a CSS által nyújtotta lehetőségekkel éltem. Így nem kell minden oldalon külön megszerkeszteni a felületet.

A CSS (Cascading Style Sheets) egy olyan leírónyelv, melynek segítségével különböző stíluslapokat hozhatunk létre és ágyazhatunk be HTML honlapjainkba. Ezek a stíluslapok befolyásolják az oldal megjelenését (design szempontjából kiváló nyelv). Meghatározhatjuk vele, hogy az egyes HTML tagek (bekezdések, táblázatok, címsorok stb.) hogyan jelenjenek meg, meghatározhatjuk a méretüket, színüket, stílusukat, típusukat stb.

A CSS dokumentumunkra forráskódban hivatkozunk:

```
<LINK href="stilus.css" type=text/css rel=stylesheet>
```



Az egyes formázásoknál még külön kell hivatkozni a CSS-ben megadott osztályokra.

```
<DIV id=vonal1></DIV>
```

A CSS elemek

Egy CSS utasításnak 2 fő része van: a *szelektor*, a *deklaráció*.

- A *szelektor* azt a HTML elemcsoportot jelöli ki, amit formázni szeretnénk (pl. P, UL, H1 stb.)

```
#vonall H1 { COLOR: #C0C0C0 }
```

- A *deklaráció* írja le a szelektorban megadott HTML tagek formázását. Ennek általában több része van (paraméter, érték), ugyanis meg kell adni, hogy mit akarunk a szelektorban formázni (pl. szín - color), utána pedig azt, hogy hogyan akarjuk formázni (pl. fehér - white).

A deklarációk csoportosításban vannak megadva az egyszerűség és a jobb átláthatóság miatt.

- A **CLASS** szelektor

Ezzel rugalmasabban tudunk stíluslapokat rendelni a dokumentumunkhoz. A lényege az, hogy a szelektorokhoz hozzárendelünk egy nevet, amire a dokumentum törzs részében bármikor hivatkozhatunk (CLASS="szelektorhoz rendelt nev"). Ez akkor jöhet nagyon jól, ha például nem minden bekezdést szeretnénk kék színűre változtatni, hanem csak egy részét.

```
P class=menukategoria
```

Példa programból:

PHP kódban a CSS megjelenése:

```
<html>
<head>
<title>Webáruház</title>
</head>
<LINK href="stilus.css" type=text/css rel=stylesheet>
<body>
<DIV id=vonal1>
<H1>Webáruház</A></H1></DIV>
<DIV id=vonal2></DIV>
<DIV id=vonal3><SPAN>.</SPAN></DIV>
<DIV id=menu_doboz>
<br>
<br>
<tr>
<UL id=menulista>
<LI class=menukategoria>Opciók
<LI><A href="katalogus.php">Vissza</A>
<br>
```

```
</tr>
</LI></UL>
</DIV>
<DIV class=tartalom>
</DIV>
</body>
</html>
```

## 6. A webáruház általános felépítése

A komplett áruház két részre osztható egy vevő és egy eladó. Szerkezetileg megjelenésbe, stílusba nincsen eltérés. Csak a funkciók mások.

### 6.1. Vevői oldal

A bejelentkező vevői oldalon lehetőség van arra, hogy megnézzük az árlistát. Mivel vannak olyan vásárlók, akik nem akarnak interneten vásárolni, áraink azonban érdeklik őket. Lehetnek offline vásárlók is.

Információs oldalon kérhetünk segítséget.

Csak azok tudnak a webáruházban vásárolni, akik belépnek. Ha nincsenek regisztrálva, ezt könnyen megtehetik. Mivel itt a vásárlást meghatározó adatokról van szó, aki nem megfelelően tölti ki a regisztrációt, annak érvénytelen a megrendelése. Rövid regisztráció után beléphet a felhasználó a webáruházba. Az árlistából kiválasztva töltheti meg virtuális kosarát, a kosár tartalma mindaddig változtatható, míg nem véglegesíti. A felhasználó módosíthatja adatait, kivéve a jelszavát, viszont regisztrációját nem tudja törölni.

### 6.2. Eladói oldal

Az eladó oldal sokkal összetettebb. A bejelentkező oldalon csak belépésre van lehetőségünk. Mint már korábban említettem, csak az a személy regisztrálhat egy másik személyt, akinek erre jogosultsága van. Az eladó oldalon van az egész portál motorja. Az adatbázist itt töltik fel és nekik mindenről tudniuk kell.

Lehetőség nyílik kategória, áru felvitelére, módosítására. Ezek segítségével tudjuk szerkeszteni az árlistát. A webáruházban megrendelt termékeket lekérdezhetik, ily módon teljesíthetők a megrendelések.

Azok, akik offline módon vásárolnak, külön felületen rögzíthetjük a számlájuk tartalmát.

## 7. Megvalósítás (kódolás)

A kódolás PHP nyelven történik, mivel egy nagyon logikus, sok lehetőséget biztosító programnyelv.

A PHP egy nyílt forráskódú webes parancsnyelv, mely csatlakozott a Perl-höz, az ASP-hez és a Java-hoz a dinamikus webes alkalmazások készítéséhez rendelkezésre álló eszközök palettáján. Amely régen túlnőtt eredeti jelentőségén. Születésekor csupán egy makro készlet volt, amely személyes honlapok karbantartására készült. Később a PHP képességei kibővültek, így egy önálló programozási nyelv született, amely képes nagyméretű Webes adatbázis-alapú alkalmazások működtetésére is.

### 7.1. Vevői oldal

#### 7.1.1. Belépés

Ahhoz be tudjunk lépni, előtte regisztrálnunk kell. Ha ezen túl vagyunk, akkor beléphetünk a portálra. Belépéskor ellenőrizni kell, hogy a felhasználó tényleg regisztrált-e. Ezek után a felhasználó azonosítója bekerül a „session\_register”-be, mely addig ott is marad, míg ki nem lép.

Az azonosítónak az online vásárlás egész menete alatt fontos szerepe van, mivel ehhez kapcsolódnak a vásárolt tételek.

Példa a jogosultság ellenőrzésére:

```
?php
session_start();
if (!isset($_SESSION["latogato"])) {
header("Location: index.php");
exit;
}
$latogato = $_SESSION["latogato"];
?>
```

### 7.1.2 Regisztráció

A regisztráció az egyik lekomplikáltabb része az oldalnak, mivel fontos, hogy minden mező ki legyen töltve és ne kerüljön fel az adatbázisba két ugyanolyan login vagy jelszavú ember.

Először a login ellenőrzése történik meg:

Példa programból:

```
if (!empty($felhlogin)) {
    $query = "SELECT felh_login FROM felhasznalo WHERE
    felh_login='$felhlogin' ";
    $result = mysql_query($query);
    if (mysql_num_rows($result)!=0) {
        $uzenet .= "<font color=red>A '$felhlogin' login név már
        foglalt, válassz másikat!</font><br>";
        $hiba = true;
        $fogl1=true;
    }
    } else {
    $uzenet .= "<font color=red>Nem adtál meg 'Login
    nevet'!</font><br>";
    $hiba = true;
    }
    <tr>
    <td>Teljes név</td>
    <td><input type="text" name="felhnev" value="<?=( $hiba ?
    $felhnev : "" )?>"><? if(empty($felhnev)&&$hiba==true) {echo
    " <font color='#CC3333' >*</font>"; } ?></td>
    </tr>
```

A jelszót két beviteli mezőbe vesszük fel és ha ezek megegyeznek csak akkor kerül be az adatbázisba, különben hibaüzenetet kapunk. Kétszeres ellenőrzés.

Példa programból:

```
if (!empty($jelszo) or !empty($jelszo2)) {
    if ($jelszo!=$jelszo2) {
        $uzenet .= "<font color=red>Nem egyezik a két
        jelszó!</font><br>";
        $hiba = true;
        $negy=true;
    }
    } else {
    $uzenet .= "<font color=red>Nem adtál meg 'Jelszó'-
    t!</font><br>";
    $hiba = true;
    }
    }
```

```

<tr>
<td>Jelszó</td>
<td><input type="password" name="jelszo"><?
if((empty($jelszo)&&$hiba==true) ||
($negy==true&&$hiba==true)) {echo " <font color='#CC3333'
>*</font>"; } ?></td>
</tr>
<tr>
<td> Jelszó (megerősít)</td>
<td><input type="password" name="jelszo2"><?
if((empty($jelszo2)&&$hiba==true) ||
($negy==true&&$hiba==true)) {echo " <font color='#CC3333'
>*</font>"; } ?></td>
</tr>

```

A többi mező, amikbe előfordulhat az egyezés, ott csak azt vizsgáljuk, hogy van-e tartalma. Bármely eltérésnél hibaüzenetet kapunk:

**„Nem adtál meg 'Város nevet!'”** és egy csillag jelenik meg a hibás mezőnél.

### 7.1.3. Vásárlás

A belépés után, ha valamilyen adatot a regisztrációkor nem jól adtunk meg, esélyünk van kijavítani, de a jelszót nem módosíthatjuk.

Az webáruházban kategória alapján kereshetjük meg az oda tartozó árut, és úgy tehetjük be kosarunkba. A kosár tartalma ugyanúgy megtalálható a megrendelések táblába, csak állapot változókkal vannak ellátva. Míg nem zárul le a kosár tartalma, az állapota „folyamatban” funkcióval rendelkezik. Ahogy véglegesítjük a megrendelést, a kosár állapota megváltozik „feldolgozva”-ra és a megrendelések állapotba „folyamatban” lesz, Ez csak akkor változik át „feldolgozva”-ra, ha a boltban az eladó teljesíti a megrendelést.

Példa programból:

```

$kosarba="INSERT into kosar
VALUES ('', '$latogato', '$_POST[aru_azon]', '$_POST[megr_menny]'
, now()) ";
mysql_query($kosarba);
$megrendeles="INSERT into megrendelesek
VALUES ('', '$latogato', '$_POST[aru_azon]', '$_POST[megr_menny]'
, now(), 'folyamatban', 'folyamatban') ";
mysql_query($megrendeles);

```

## 7.2. Eladói oldal

### 7.2.1. Belépés

Az eladói oldalon nem tudja külsős regisztrálni magát. Csak egy olyan teheti meg akinek, már van jogosultsága a rendszerhez. Talán meglehet bízni annyira az alkalmazottakban, hogy nem adják külsős kezére.

### 7.2.2. Regisztráció

A regisztráció ugyanolyan elven működik, mint a felhasználónál, csak kevesebb adat kerül rögzítésre.

### 7.2.3. Kategória szerkesztése

A PHPMyAdmin-ba nem kell felvinni, az árukat mivel erre van egy külön felület. Mikor egy új árlistát szerkesztünk, akkor először mindig a kategóriát hozzuk létre, és utána kerülnek bele az áruk.

Lehetőségünk van, kategóriát létrehozni, szerkeszteni és törölni. Fontos hozzátenni, ha egy kategóriát törölünk, akkor a hozzátartozó árukat is meg kell semmisíteni, mert azok csak elvesznének az adatbázisban. A felvitt kategóriák azonnal meg is jelennek táblázatos formában.

Példa programból:

```
$lek = "SELECT * FROM categoria ORDER BY azon DESC";
echo "<table width='80%' border='1'> \n";
echo "<tr><td><b>Kategória név</b></td><td><b>Kategória
leírás</b></td></tr> \n";
$redm = mysql_query($lek);
while ($sor = mysql_fetch_array($redm)) {
echo "<tr><td>$sor[1]</td><td>$sor[2]</td></tr> \n";
```

### 7.2.4. Áru szerkesztése

Az áru adatait egyszerű és gyors módon tölthetjük fel. A művelet elején ki kell választani, melyik kategóriába szeretnénk betenni az árut és utána lehetőségünk nyílik megadni az adatokat.

Példa programból:

Válassz kategóriát:

```
<select name="kateg">
```

```

<option value="0">---Válassz kategóriát---</option>
<?php
$kapcs=mysql_connect(DBHOSTNAME,DBUSERNAME,DBPASSWORD) or
die(mysql_error(hiba));
mysql_select_db(DBNAME,$kapcs) or die(mysql_error());
$lek = "SELECT * FROM categoria";
$eredm = mysql_query($lek);
while ($sor = mysql_fetch_array($eredm) ) {
?>
<option value="<?php echo $sor["azon"]; ?>"><?php echo
$sor["kat_nev"]; ?></option>
<?php
}
mysql_close($kapcs);
?>
<br>
</select>

```

Természetesen itt megtehetjük, hogy bármikor megváltoztathatjuk az egyes áruk adatait. Árváltozás estén, vagy ha már nincsen az adott termékből, töröljük.

Példa programból:

```

echo"<form action='modositaru.php'>\n";
$arunev= str_replace("<br />", "", $adat[1]);
$aruleir= str_replace("<br />", "", $adat[2]);
$arunettoar= str_replace("<br />", "", $adat[3]);
echo"<input type='text' name='arunev' value='$arunev'>";
echo"<input type='text' name='arunettoar'
value='$arunettoar'>";
echo"<textarea name='aruleir' rows='10'
cols='30'>$aruleir</textarea>";
echo "<input type='hidden' name='aruazon' value='$adat[0]'>
\n";
echo "<input type='submit' value='Módosít'> \n";
echo "</form> ";

```

### 7.2.5. Megrendelések

Az új megrendelések úgy jelennek meg az eladó oldalon, hogy a vevők egy állapottal vannak ellátva: „van” vagy „nincs”. Ha a felhasználó véglegesíti a kosarát, akkor a hozzá tartozó állapot átvált a „van” értékre, és ezáltal megjelenik az eladónál. A „van” akkor lesz újra „nincs”, ha az eladó teljesíti a megrendelést.

Példa programból:

```

$hozzad="UPDATE megrendelesek SET állapotmegr='feldolgozva'
WHERE állapotmegr='folyamatban'and felh_azon='$vevo'";

```

```
mysql_query($hozzad);  
$van="update felhasznalo set vette='nincs' where  
azon='$vevo'";  
mysql_query($van);
```

### 7.2.6. Számlázás

A számlázás az offline vásárlók adatait rögzíti. Hasonló eljárási módszerben, mint az online vevőknél csak itt kevesebb adatot kezelünk, már csak azért is mert a fizetés formája készpénzes.

Minden egyes vásárlás kap egy azonosítót, melyet számlaszámmak is tekinthetünk. Lekérdezésnél a számlaszámra kattintva lehívhatjuk a tételes számlát.

## 8. Üzembe helyezés, tesztelés

Az oldal kivitelezés alatt folyamatos tesztelésnek volt kitéve, hogy egyes hibák ki legyenek szűrve. Törekedtem arra, hogy bármely felmerülő hiba esetén kapásból megtudjuk, hol van annak a forrása. Az egyes hibák meghatározására és kiírására a nagy segítséget jelentett a PHP néhány beépített függvénye. A „die” és a „mysql\_error”, amelyek az adatbázishoz kapcsolódás során keletkező hibát a megadott sztring segítségével közlik. például: *die(Hiba az adatbázisban!)* .

Egyik legfontosabb adatot, a jogosultságot a SESSION-be tároltam el, és ennek meglétének ellenőrzésére felhasználtam a „print session\_encode();” függvényt, mely kiírja az abban a pillanatban lévő SESSION tartalmát.

A problémák feltárásának egyik legjobb módszere, ha a fontosabb műveleteket if-else vizsgálatnak rendeljük alá. Egyszerű és bármely hiba esetén, pontosan meg tudjuk határozni a probléma forrását.

Mikor felveszek az adatbázisba egy adatot, hogy meggyőződjek arról, hogy sikeres-e a művelet, az állapotnak megfelelő üzenetet iratok. Ugyanez van a regisztrációs folyamatnál is. Ahol valami nem egyezik meg, vagy hibás, ott hibüzenetet kapunk. De emellett a PHPMyAdmin-ba is figyelemmel kísérhettem, hogy bekerült-e az adatbázisba.

A rendszer tesztelése során alkalmaztam statikus és dinamikus módszereket is.



Statikus tesztelés esetén csak a programkódot vizsgáljuk, ami a programozás során folyamatosan megtörtént. Mivel a helyes működést ilyenkor mindig láttam, illetve a PHP által adott hibaüzenetek kijavítása után minimálisra csökkenthettem a programban lévő, rosszul megírt parancsokat.

Dinamikusan is tesztelve lett a szoftver. A HTML-ben való honlap készítésekhöz hasonlóan itt is azonnal ellenőrzésre kerültek az adott oldalak. Az ellenőrzés olyan folyamat, hogy ha nem megy valami, akkor addig kell próbálkozni az egyes működési módokkal, hogy futóképes programot kapjunk.

A dinamikus tesztelés során, a *fekete és fehér doboz* módszereket alkalmaztam. A fekete doboz módszer során nem a program belső szerkezetét tartjuk elsődlegesnek, hanem különböző feladat meghatározásokkal választjuk meg a teszteseteket.

A vevő oldalon leteszteltem egy vásárlási folyamatot. Hogyan kerül az áru a kosárba, majd megvásárlásra. Ha módosítok az adataimon, akkor azok jelenjenek meg a képernyőn. Ugyanúgy az eladói oldalon, figyelem hogy a feladott megrendelés, tényleg megjelenik-e. Ha lezárom a megrendelést, akkor az tényleg „lezáródik”-e. A számla készítésénél, amit felviszek a számlába, annak meg kell jelennie. Ezeknek használatával meggyőződhettem a program helyes működéséről. A hibák azonnal megjelentek, ha voltak.

A fehér doboz módszerek alkalmazása során is, a fekete dobozhoz hasonlóan, beléptem mindkét felhasználói oldalon és teszteltem az egyes funkciókat, párhuzamosan a forráskód ellenőrzésével. Két lehetőséget vizsgáltam Vagy működik, vagy nem. Ha nem megy, akkor mit ír ki és ezt a hibát hogyan lehet orvosolni.

Egyik problémám volt, hogy az egyes azonosítókat nem mindig tudtam jól betenni a SESSION-be, vagy nem igazán akarta átvinni egyik oldalról a másikra. A `<form action="" method=""></form>` használtam az egyes adatok átviteléhez. Ehhez tartoznak a GET és a POST metódusok, melyek segítségével lekérhettem az adatokat. Olyankor, ha adatokat küldtem át másik oldalakra, annak a lapnak az elejére, ahová a beviteli mezők értékei érkeznek, kiírtam ezeket a változókat. Így láttam hogy, megérkeznek-e az adatok. Ha ezek elküldése sikeres volt, akkor kezdtem csak hozzá az adattáblába való beillesztés leprogramozásához.

Külön figyelmet fordítottam arra, hogy a menükben és az egyes funkciókban szereplő linkek helyes oldalakra mutassanak. Mivel elég sok forrásfájllal dolgoztam, néha előfordult, hogy egy-két helyre rossz link került. Ezek észrevétele és javítása is része a fekete és fehér doboz tesztelési módszereknek.

## 9. Összefoglalás

Az elkészült portál az egyszerű, kényelmes online vásárlásra szolgál. Teljes mértékben web-alapú, ami azt jelenti, hogy mint a vevő, mint az eladó bármikor el tudja érni az interneten.

Egyre több háztartásba jut el az internet és mint tudjuk az ember nagyon kényelmes. Talán ez egyik oka, hogy növekszik az érdeklődés az e-kereskedelelem iránt. Nem kell napokat tölteni azzal, hogy járjuk a boltokat, hanem leülünk a számítógép elé és pillanatok alatt kiválaszthatjuk a számunkra megfelelő terméket.

Sokan félnek az internetes vásárlástól, mert sok adatot adnak meg magukról és pénzmozgás is történik. Nem csoda mikor nap, mint nap azt lehet hallani, hogy mennyi vírus, internetes támadás és megtévesztés áldozataivá válnak sokan. Talán ha biztonságosabbnak éreznék a világhálót, akkor sokkal többen élnének az e-kereskedelelem által nyújtott lehetőségekkel.

### Futási környezet

Sokan használnak régebbi gépeket internetezésre és sok probléma szokott adódni az egyes flash-eknél, hogy nem képes a megjelenítésre. Ezért, törekedtem arra, hogy a lehető legegyszerűbb gépen is lehessen látogatni az oldalt fennakadások nélkül.

### További fejlesztések

A honlap nem tartalmazza a raktár adatait. Egy olyan tábla kellene még hozzá, mely a raktáron lévő eszközök mennyiségét, és egyéb adatait tartalmazza. Ezt össze lehetne hangolni a vásárlásokkal, a készletmozgásokkal és a könyveléssel.

A másik fontos fejleszteni való, az egyes megrendelések külön-külön való lementése egy dokumentumba, egy nyomtatható formába.

## A honlap megtalálható az interneten.

Vevői oldal

<http://vargacsaba.uw.hu/index.php>

Login: meron

Jelszó: codename

Eladói oldal

<http://vargacsaba.uw.hu/elado.php>

Login: meron

Jelszó: codename

## 10. Felhasznált irodalom

- [1] Meloni, Juli C - A PHP, a MySQL és az Apache használata  
Budapest Panem, 2004
- [2] Schlossnagle George - PHP fejlesztés felsőfokon  
Budapest Kiskapu, 2004
- [3] Wyke, R. Allen ; Walker, Michael J. ; Cox, Robert M.- PHP fejlesztők kézikönyve  
Budapest Kossuth cop, 2002
- [4] Moulding, Peter - PHP haladóknak  
Budapest Perfact-Pro, 2002
- [5] Reese, George - A MySQL kezelése és használata  
Budapest Kossuth cop., 2003
- [6] Livingston, Dan - CSS és DHTML webfejlesztőknek  
Budapest Kossuth cop., 2003
- [7] Paczona Zoltán - HTML technikák a gyakorlatban  
Budapest Computer Panoráma cop., 2002
- [8] <http://informatika.bke.hu/root/Project/telepiac.nsf/0/5f13276f8caf4152c12569d6003da40c?OpenDocument#1.%20Bevezet%C3%A9s%20%C3%A9>

## Mellékletek

- 1 db CD

A CD tartalma:

- Webáruház adatbázisa
- PHP oldalak
- Elérés beállítása