

SZAKDOLGOZAT

Hajdu Szabolcs

Debrecen

2010

Debreceni Egyetem
Informatikai Kar

EGY INTERNETES VÁSÁRLÁS IMPLEMENTÁLÁSA

Témavezető:

Mecsei Zoltán

egyetemi tanársegéd

Készítette:

Hajdu Szabolcs

gazdaságinformatikus

Debrecen

2010

TARTALOMJEGYZÉK

1. Bevezetés.....	4
2. Az elektronikus kereskedelemről általában.....	6
2.1. Az elektronikus kereskedelem legfontosabb előnyei.....	6
2.2. Az elektronikus kereskedelem legfontosabb hátrányai.....	7
2.3. A hagyományos és az e-kereskedelem összeolvadása.....	8
3. Online vásárlási szokások – kérdőív elemzése.....	9
4. Felhasznált technológiák.....	13
4.1. HTML és XHTML.....	13
4.2. CSS és a design.....	16
4.3. PHP, Apache és MySQL.....	19
4.4. Keresőoptimalizálás.....	22
4.4.1. Onsite optimalizáció.....	23
4.4.2. Offsite optimalizáció.....	24
5. A webáruház megvalósítása.....	26
5.1. Az adatbázis tervezése és elkészítése.....	26
5.1.1. Előzetes táblaszerkezet meghatározása.....	26
5.1.2. A normalizálás folyamata.....	27
5.1.3. A végleges adatbázis felépítése.....	27
5.2. Felhasználói felület létrehozása.....	30
5.2.1. Regisztráció és belépés.....	31
5.2.2. A kosárba helyezés művelete.....	33
5.2.3. A rendelés folyamata.....	34
5.3. Adminisztrációs felület létrehozása.....	35
6. Az oldal tesztelése.....	38
7. Összefoglalás.....	40
Irodalomjegyzék.....	42
Függelék.....	44
Köszönetnyilvánítás.....	50

1. Bevezetés

Az internet kialakulásának kezdetén még nem is sejtették, hogy mekkora dolgot találtak fel, és hogy ezzel a találmánnyal az egész emberiség életét meg fogják tudni változtatni. Az „ősidőkben” – amikor még csak kezdetlegesen létezett az internet – az atomfizikusok a pusztán szöveges információk mellett képekre, ábrákra és formázott szövegekre vágytak. Így kezdődött el egy olyan folyamat, mely a weboldalakat egyre szebbé és egyre inkább interaktívvá varázsolta.

Az akkori internet felhasználói főként tudósok voltak. A weboldalak viszont az akkori használatuknál sokkal többre voltak hivatottak. Az idő múlásával a vállalkozó kedvű cégek felismerték a benne rejlő hatalmas lehetőségeket. Segítségével a világ tulajdonképpen összezsugorodott és az információ áramlása hirtelen felgyorsult. Ma már teljesen mindegy, hogy az ország melyik részén élünk, ugyanis ha rendelkezünk internetkapcsolattal, szinte bárhol, bárkivel kapcsolatba léphetünk, kikapcsolódhatunk, blogolhatunk, tárgyalásainkat intézhetjük és akár vásárolhatunk is. Mivel a legtöbb vállalat profitorientált, ezért a hagyományos marketing eszközök mellett bevetették az internetmarketing erejét is. Eleinte még csak cégük reklámozására használták, később azonban rájöttek, hogy egyszerűen és viszonylag olcsón növelhetik vásárlóik számát. Ehhez mindössze arra volt szükség, hogy megnyissák boltjukat az internet hatalmas piacán is. Manapság már szinte elképzelhetetlen, hogy egy ismertebb boltnak, áruháznak, vagy kisebb-nagyobb vállalatnak ne legyen internetes elérhetősége, vagy akár webáruháza.

Gazdaságinformatikusként nagyon érdekel ennek a folyamatnak mind a közgazdasági, mind pedig az informatikai vonala. Szakdolgozatomban is szeretném ötvözni ezt a két irányt és megmutatni, hogy mostanában mennyire összefonódott ez a két alapjaiban különböző szakma. Kérdőívem segítségével szeretnék képet kapni az emberek internetes vásárlási szokásairól, mint például, hogy milyen terméket vásárolnak webes áruházakból, milyen fizetési módot részesítenek előnyben, és hogy ezek hogyan oszlanak meg nemek, foglalkozás, vagy éppen kor szerint.

A kialakult kép alapján pedig szeretnék megtervezni és utána megvalósítani egy olyan webshopot, amely figyelembe veszi ezeket a vásárlási szokásokat és olyan igényeket elégít ki, melyeket esetlegesen más cégek még nem fedeztek fel, vagy úgy gondolják, hogy nincs rá szükség. Ezen kívül szeretném, hogy ne csak a funkciók legyenek hasznosak, hanem a kényelem szempontjából is megfelelően legyen kialakítva az oldal. Így tehát nem csak közgazdasági és programozási ismereteimet szeretném felhasználni a szakdolgozatban, hanem csekély webdesign és webergonómiai ismereteimet is.

Bízom benne, hogy egy olyan oldalt fogok tudni létrehozni, ami nem ugyanarra a sablonra épül, mint a legtöbb, ma használatban lévő webáruház.

2. Az elektronikus kereskedelemről általában

Az utóbbi idők egyik leggyakrabban használt kifejezése az elektronikus kereskedelem, más néven az e-kereskedelem, vagy e-commerce. Ez a kifejezés jelentős szerepet játszik napjaink üzleti kereskedelmében és az informatikában egyaránt, és először az interneten való web alapú vásárlások elnevezésére használták.

Az első meghatározások közös jellemzői az alábbiak voltak:

- Kizárólag a kiskereskedelmi tevékenységre, azon belül is a vásárlót érintő olyan szakaszokra vonatkoztatta az elektronikus kereskedelmet, mint az áru/szolgáltatás hirdetésére és a megrendelésére.
- A vásárló, mint természetes személy és a termékek vagy szolgáltatások értékesítését végző vállalatot képviselő rendszer közti kapcsolatra terjedt ki.
- Informatikai eszközrendszerét tekintve az internetes web elérésére korlátozódott.

Az interneten való vásárlásoknak köszönhetően jelentősen átalakult a piac fogalma. A világháló is tulajdonképpen egy hatalmas piacterré alakult. Mind az eladóknak, mind a vevőknek több ponton jelentett előnyt és egyben hátrányt is ez az átalakulás.

2.1. Az elektronikus kereskedelem legfontosabb előnyei:

- *Az eladók könnyen követhetik a terméküket*, hiszen nagyon sok dolgot le tudnak szűrni a vásárlási szokásokból, illetve a vevők visszajelzéseiből. Ha például egy termék nem hoz elég bevételt, akkor könnyebbé válik ennek felismerése.
- *A piacnak nincsenek határai, és időbeli korlátai*, hiszen egy weboldalt, vagy egy webáruházat a világ bármely pontján, a nap 24 órájában megtekinthetünk, ha rendelkezünk internetes kapcsolattal.
- *Nagyobb és élesebb verseny* alakul ki a vállalatok között, hiszen bármelyik cég, bármikor megtekintheti a konkurens cég árait, ezáltal úgy alakítja versenypolitikáját, hogy az minél kedvezőbb legyen a vevő számára.

- Az eladók *friss, és több információval* rendelkeznek a vásárlókról, hiszen rengeteg dolgot tudnak meg a felhasználóról csak a regisztráció útján. Ezen kívül a vásárlási szokásaikról is teljes képet kaphatnak, ha megnézik, ki, mikor, mit vásárolt.
- A fogyasztók számára *egy webshop könnyebben hozzáférhető*, mint a hagyományos piactér. Az interaktív felületeknek és katalógusoknak köszönhetően az adott termékről mindent megtudhatunk gyorsan és közvetlen ráfordítás nélkül.
- A vevő *teljes kényelemben* helyezheti magát, hiszen otthonról, a székből történik a vásárlás.
- A rengeteg animáció és a szemet gyönyörködtető effektek, valamint a honlap kinézete sokkal *élvezetesebbé teszi a vásárlást*. A vásárló egyben élményekkel is gazdagodik a termék megvásárlásán kívül.

2.2. Az elektronikus kereskedelem legfontosabb hátrányai:

- Ahhoz, hogy egy webáruház megfelelően működjön, viszonylag *nagy befektetésre* van szükség. Nagyon fontos a marketing-módszerek megfelelő alkalmazása, és a keresőoptimalizálás. Egy új oldal „felfuttatásához” rengeteg pénz, idő és tudás szükséges.
- A legtöbb vevő a mai napig azért nem vásárol olyan szívesen webáruházban, mert *fél, hogy nem azt fogja kapni, amit ígérnek neki*. Nem tudja megfogni és megnézni élőben a terméket, bizonytalan lesz a termék minőségében.
- A termék kifizetésére többféle módszer létezik manapság, ennek ellenére sokszor előfordul, hogy *a vevő nem kapja meg a terméket*, hiába fizetett érte. Ez nem mindig az üzemeltető hibája, sokszor maga a fizetési rendszer nem működik megfelelően.
- Végül, de nem utolsó sorban az egyik nagyon fontos hátrány, a *személyes érintkezés hiánya*, ami a vevőben bizalmatlanságot kelthet és inkább csak információszerzésre használja az online boltokat, vagy marad a hagyományos vásárlásnál. Éppen ezért nagyon fontos, hogy különböző módszerekkel biztosítsuk a vásárló bizalmát.

2.3. A hagyományos és az e-kereskedelem összeolvadása

Újabban egyre elterjedtebbé válik az úgynevezett „*click and mortar modell*”, vagy más néven *hibrid vállalatok*, ami azt jelenti, hogy a hagyományos kereskedelmi vállalatok belépnek az e-kereskedelembé, és ügyleteiket részben elektronizálják, vagy önálló online üzletágot hoznak létre.

Ilyen üzleti modellt alkalmaz például az angliai Tesco. Amennyiben egyik vásárlója az interneten megrendeli valamely termékét, vagy termékeit, akkor azt a Tesco ingyenesen házhoz szállítja.

A modell nagy előnye, hogy a kereskedő cégét már ismerik az emberek, mindössze kiterjeszti szolgáltatásait. A vevő azonnal megkaphatja a terméket és a vállalat személyesen is tudja tartani a kapcsolatot a vevőivel. Amennyiben hibás terméket szállítanak házhoz, abban az esetben a vevő akár személyesen is vissza tudja vinni, hiszen a vállalat nem csak online létezik.

Véleményem szerint nem szükséges elválasztani a hagyományos és az e-kereskedelmet. Mind a kettőnek megvannak a maga előnyei és hátrányai. Ahhoz, hogy egy „majdnem tökéletes” kereskedelmi vállalatot hozzunk létre, ideális kiindulás lehet a hibrid modell, hiszen itt ötvözve vannak a kétfajta vállalkozás előnyei és a vásárló szabadon eldöntheti, hogy milyen formában szeretné megvásárolni az adott termékeket.

3. Online vásárlási szokások – kérdőív elemzése

2010 tavaszán készítettem egy kérdőívet, melyet feltöltöttem a saját tárhelyemre és a nagyobb közösségi oldalakon közzétettem az oldal linkjét. A kérdőív célja az volt, hogy a kérdésekre adott válaszok alapján egy összképet kaphassak a mai kor online vásárlási szokásairól. A kitöltése teljesen önkéntes és névtelen volt, mindössze egy e-mail címet kellett megadnia a válaszolónak, de ez csak azt a célt szolgálta, hogy egy ember csak egyszer tölthesse ki. Összesen 181 ember válaszolt a kérdésekre, tehát a felmérés nem mondható reprezentatívnak, de kisebb következtetéseket ezekből is le lehet vonni.

A válaszolók 64 százaléka nő volt, míg 36 százalékuk férfi (*1. ábra*), életkoruk pedig 10 és 58 év közé esik. Csak ebből a néhány adatból is láthatjuk, hogy rendkívül széles skálán mozog az interneten böngészők száma.

A megkérdezettek 72 százaléka 16 és 25 év közötti diák (*2. és 3. ábra*), amiből viszont az következik, hogy még mindig a fiatalok azok, akik a legtöbbet használják az internetet. Még mindig nem mondhatjuk teljesen azt, hogy minden korosztályt el tudunk érni az interneten, tehát ha kifejezetten idősebbeknek szóló termékeket szeretnénk árulni, akkor nem biztos, hogy a legjobb módszer az internetes áruház létrehozása. Ahhoz, hogy az idősebb korosztályt is bevonhassuk a netes kereskedelembé, még el kell telnie pár évtizednek.

Arra a kérdésre, hogy *milyen gyakran vásárolnak az interneten keresztül*, viszonylag vegyes válaszok érkeztek. A válaszadók 61 százalékáról azt mondhatjuk, hogy ritkán keresnek fel webáruházakat vásárlás szempontjából. Sőt, 72 százalékuk még soha nem vásárolt online, vagy inkább csak böngész a webshopokat, de nem vásárol. Ezek azok az emberek, akik nem igazán bíznak a webáruházakban és inkább elsétálnak a legközelebbi boltokba és megveszik ott a kiszemelt terméket, maximum információszerezés miatt nézelődik a weblapokon. Jelenleg a hibrid vállalatok azok, akik ezt a vevőkört elcsábíthatják, hiszen ezek a cégek rendelkeznek tényleges üzletekkel és termékeik reklámozását alacsonyabb költséggel, saját honlapjukon tudják megoldani. Extra szolgáltatásaikkal tudják igazán magukhoz vonzani őket, mint például az ingyenes házhozszállítás.

Azon válaszolók aránya, akik rendszeresen felkeresik az online üzleteket és vásárolnak is, elenyésző, mindössze 18 százalék. Ezen adatok alapján elmondható, hogy Magyarországon még gyerekcipőben jár az e-kereskedelem és még el kell telnie pár évnek, hogy a vásárlókban kialakuljon egyfajta bizalom. (5. ábra)

A következő kérdés azokhoz szólt igazán, akik már vásároltak interneten keresztül. *Milyen termékeket vásároltál eddig online áruházakban?* Kiemelkedően magas volt a könyv, illetve a ruházat és sportfelszerelés iránt való érdeklődés. A könyv iránti magas kereslet egyáltalán nem meglepő, hiszen ez az a termék, aminél egyáltalán nem fontos a személyes megjelenés a boltban, valamint az sem fontos, hogy lássuk, vagy megfogjuk, mielőtt megvesszük. Mindent, amit tudnunk kell egy könyvről, azt a tájékoztatóban elolvashatjuk. Erre nagyon sok vállalkozó ráébredt, éppen ezért a legtöbb könyvesbolt már rendelkezik webáruházzal, ahol a vásárlók otthonról, kényelmesen kereshetnek a virtuális polcokon.

Talán a legjobb példa az internetes könyváruházra a www.amazon.com. A hagyományos könyveken kívül árulnak elektronikus könyveket, úgynevezett e-bookokat, amelyek napjainkra egyre inkább elterjedté válnak. Az Amazon nem rendelkezik igazi könyvesbolttal, csak egy interaktív és rendkívül letisztult weblappal. A regisztráció során az emberek válaszolhatnak olyan kérdésekre, amelyek az érdeklődési körére vonatkoznak és ezek alapján, valamint több apró trükknek köszönhetően a belépett felhasználó személyre szabott ajánlatok sorát tekintheti meg. Az űrlap kitöltésén kívül az oldal figyeli a megvásárolt termékeket is, így még pontosabban meg tudja határozni a vásárló szokásait, érdeklődését.

A ruházat és a sportfelszerelés már kissé meglepőbb lehet, hiszen a legtöbben szeretnék megfogni a ruha anyagát, szeretnék felpróbálni, hogy biztosan jól áll-e neki az adott ruhadarab. Ennek ellenére mégis egyre több olyan e-bolttal lehet találkozni, ahol ruhákat vásárolhatunk. Ennek oka feltehetőleg az egyediség. A legtöbb fiatal szeret egyedien öltözködni és újabban egyre elterjedtebb az a funkció az online ruhaüzletekben, hogy a vásárló teljesen egyedivé varázsolhatja pólóját, nadrágját. Ezen felül vannak olyan ruházati cikkek, kiegészítők, amiket csak nagyon ritkán lehet kapni a hagyományos boltokban, ezért az interneten való megvásárlásuk sokkal egyszerűbb és olcsóbb.

Az előbbire egy nagyon jó példa a www.polomania.hu. Saját tervezésű pólót készíthetünk az oldal felületén, majd az elkészített „művet” meg is rendelhetjük. Kívánságra házhoz is szállítják. (6. ábra)

A hatodik kérdésem az volt, hogy *milyen fizetési módot részesít előnyben az internetes vásárlások alkalmával*. A válaszadók 31 százaléka, ha interneten vásárol, akkor a termék kézhezvétele után szeretne fizetni. Ez is a webboltok iránti bizalmatlanságot támasztja alá. Nagyon sokan nem szeretnek előre fizetni, és ha nincs ilyen lehetőség, akkor inkább nem is vállalják a kockázatot, elállnak a vásárlástól.

A modernebb fizetési rendszereket, mint például a PayPal, vagy a bankkártyás és hitelkártyás fizetések, viszonylag kevesen ismerik, vagy ha ismerik is, nem szívesen használják, mert félnek attól, hogy ha előre átutalják a pénzt, akkor nem, vagy csak késve kapják kézhez a vásárolt terméket. Saját tapasztalatból mondhatom, hogy ez manapság már egyre ritkábban fordul elő és szerintem lassan az ügyfelek is rá fognak jönni, hogy nekik is jobban megéri, hiszen több vállalkozás is plusz díjat számol fel azért, ha valaki utólag kíván fizetni. (7. ábra)

Azon kérdésem, hogy *vásárláskor a hazai vagy a külföldi oldalakat részesítik előnyben*, az előzőekre kapott válaszok alapján nem igazán hozott meglepetést. A megkérdezettek majdnem 60 százaléka inkább a magyarországi webáruházakat böngészgeti. A külföldi oldalakra mindössze 2 százaléka szavazott. Teljesen egyértelmű, hogy ha az emberek nagy része az itthoni webshopokban sem bízik, akkor a külföldiben még úgysem fog. Rengeteg átverésről és hibás teljesítésről szóló híreket lehet hallani ismerősöktől, fórumokon, blogokon. Ma Magyarországon, aki külföldről rendel valamilyen terméket, az nagy valószínűséggel azért teszi, mert itthon nem lehet kapni. Ez teljesen logikus, hiszen feltehetőleg a szállítási költség miatt jóval drágább egy osztrák, német, angol vagy amerikai áruházból rendelni. (8. ábra)

Átlagban mennyi pénzt költesz egy hónapban online vásárlásra? A válaszadók több mint fele 5 ezer forint alatt vásárol átlagosan. Ennél nagyobb értékben mindössze a negyedik vásárol, de közülük is a nagy részük az 5 és 15 ezer forint közötti sávban. Elenyésző azon emberek száma, akik ennél nagyobb értékű dolgokat vásárolnak interneten keresztül. Ebből igazából csak azt szűrhetjük le, hogy nem szívesen adnak ki az emberek

nagyobb összegeket egy webáruházban történő vásárláskor. Viszont jelen van az a tényező is, hogy nagyobb értékű dolgokat csak ritkábban veszünk. Azok, akik akár 100 ezer forint körül, vagy afelett vásárolnak havonta, főként informatikai eszközöket rendelnek, mint például számítógép, egyéb hardverek, szoftverek, valamint mobiltelefont. Életkorukat tekintve is már a 30 év körüliekhez tartoznak. (9. ábra)

Az utolsó két kérdésem pedig arról faggatta a válaszolókat, hogy *számukra milyen előnyökkel és hátrányokkal bír egy online webáruház*. Összességében elmondhatom, hogy a legtöbben előnyként hozták fel a kényelmet, a gyorsaságot, a kedvezőbb árakat, a szélesebb választékot, a termékről több információhoz jutottak, valamint olyan dolgokat tudnak megvenni az interneten, amiket más boltokban nem lehet kapni. A legfontosabb hátrányok közé pedig a személytelenséget, az élőben való megtekintés hiányát és a biztonság hiányát sorolták. Ezeket a hátrányokat ki lehetne küszöbölni azzal, ha a lehető legtöbb vállalat arra törekedne, hogy elnyerje a vásárló bizalmát, és amit ígér, azt teljesíti is időben és a szerződés alapján.

A kérdőív elemzése során arra a megállapításra jutottam, hogy egy megfelelően megszerkesztett, biztonságos webáruház létrehozásával, igenis meg lehet nyerni a vásárlók bizalmát és Magyarországon is vonzóvá lehet tenni az e-kereskedelmet. Éppen ezért szerettem volna létrehozni egy olyan oldalt, amely a saját tudásom alapján a lehető legjobbat kínálja a vásárlóknak. Ezen munkámat szeretném bemutatni, a kezdeti lépésektől, egészen a projekt befejezéséig.

4. Felhasznált technológiák

Az elkövetkező néhány bekezdésben szeretném ismertetni a dolgozatom során felhasznált programnyelveket, technológiákat, melyekkel létrehoztam az adott webáruházat. A kódot az Adobe Dreamweaver nevű szoftverrel írtam meg. Internet Explorer 8, Firefox 3.6.3, Opera 10.50, Safari 4 és a Google Chrome 5-ös verziójú böngészőkben teszteltem az oldalakat.

4.1. HTML és XHTML

A HTML mozaikszó a *HyperText Markup Language* (*hiperszöveges leírónyelv*) rövidítése, melyet 1990-ben dolgoztak ki a World Wide Web megtervezése után 1 évvel. A jelenlegi legelterjedtebb verzió a HTML 4.0, de már készülóban van a HTML 5, mely verzió sokak szerint a Flash leváltására fog szolgálni. A HTML dokumentum egy szöveges file, de a szövegen kívül címkéket, úgynevezett tageket tartalmaz. Ezek a tagek szolgálnak a szöveg szerkezeti felépítésének, valamint megjelenésének a meghatározására. Szerkezetileg egy szövegrész lehet például címsor, lista, bekezdés, stb. Míg a megjelenésen a szövegben használt betűtípusokat, betűméreteket értjük.

Egy egyszerű HTML példa:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html>
  <head>
    <title>Hello World - HTML</title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

Az XHTML nem más, mint *eXtensible HyperText Markup Language* (bővíthető hiperszöveges leírónyelv), amelyet a HTML-ből fejlesztettek ki. A bővíthetőség azt jelenti, hogy olyan modulokat építhetünk a kódjainkba, melyekkel képesek lehetünk például bonyolultabb matematikai számítások elvégzésére vagy képek rajzolására. Az XHTML dokumentumok XML alapúak és tulajdonképpen arra lettek tervezve, hogy együttműködjenek az XML alkalmazásokkal. A webáruház elkészítése előtt azért választottam az XHTML nyelvet, mert jelenleg ez a legújabb, W3C által is ajánlott HTML szabvány. Felépítésében nagyon hasonlít a HTML-re, de szintaktikailag valamivel több megkötés található az XHTML nyelvben.

Egy egyszerű XHTML példa:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en" lang="en">
  <head>
    <title> Hello World - XHTML </title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

HTML és XHTML közötti különbségek

- Minden elemet mindig le kell zárni, valamint egymásba ágyazásnál ügyelnünk kell a sorrendre, vagyis az elemek megnyitásának fordított sorrendjében kell elhelyezni a záró elemeket.

Helytelen:

```
<b><u>Vastag, aláhúzott szöveg</b></u>
```

Helyes:

```
<b><u>Vastag, aláhúzott szöveg</u></b>
```

- Az XHTML-hez szükséges csatolnunk úgynevezett DOCTYPE meghatározást, mely tartalmazza, hogy milyen dokumentumtípus-definícióval (DTD) rendelkezik az adott oldalunk. Ugyanis ahhoz, hogy egy XHTML dokumentum érvényes legyen, egyik szükséges feltétel, hogy rendelkezzen megfelelő DTD-vel, ami meghatározza, hogy milyen szabályok szerint íródott az oldalunk. Jelenleg a W3C webhelyén található DTD verziók azok, amelyek a leginkább elterjedtek.

Az XHTML-hez három lehetséges DOCTYPE-meghatározás tartozhat:

- Szigorú: csak a szerkezeti felépítésre vonatkozó elemeket és jellemzőket használhatjuk, a megjelenítésre vonatkozókat nem.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

- Átmeneti: használható néhány megjelenítést szabályozó elem és jellemző.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
```

- Keretkészletes: kereteket tartalmazó weblapoknál használjuk.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
frameset.dtd">
```

Ha biztosak szeretnénk benne lenni, hogy valóban érvényes XHTML kódot írtunk, akkor a validátor nagy segítségünkre lehet, amit a www.w3c.org oldalon meg is találhatunk.

A W3C ezen kívül még ajánlja, hogy a DOCTYPE-meghatározás előtt adjunk meg egy XML meghatározást is, utána pedig a kezdő <html> taget kombináljuk az

XML névtérrel. Tehát a teljes ajánlás a fenti XHTML példa első három sorában látható.

- Minden elemet és attribútumot kisbetűvel kell írni.
- Az attribútumok értékeit egyes vagy kettős idézőjelek közé kell tenni.
- Minden jellemzőnek egyértelműen kell megadni az értékét, azaz nem szabad őket rövidíteni.

Helytelen:

```
<input type="radio" value="male" checked />
```

Helyes:

```
<input type="radio" value="male" checked="checked" />
```

4.2. CSS és a design

A CSS az angol *Cascading Style Sheets* rövidítése, ami egymásba ágyazott stíluslapokat jelent. A HTML által generált oldalak megjelenítését befolyásoló programozási előírás. Segítségével a weblapunk kinézetét és számos más tulajdonságát határozhatjuk meg. Befolyásolhatjuk vele, hogy hol jelenjenek meg a HTML elemek, illetve azok színét, méretét, margóját, stb. Az egymásba ágyazhatóság tulajdonképpen azt jelenti, hogy egyszerre több stíluslapot is megadhatunk és egyszerre több elemre is megadhatjuk ugyanazt a stílust. Mindig az a stílus-szabály lesz érvényes egy elemre, amelyik a hierarchiában utoljára található meg, tehát nagy szerepet kap az öröklődés a szabályok definiálásakor.

Stílusokat definiálhatunk közvetlenül a HTML dokumentumban is, de a legjobb megoldás, ha külön állományban hozzuk létre a CSS dokumentumunkat, majd azt importáljuk be a HTML kódunkba. Ekkor figyelhetjük meg a CSS egyik legnagyobb előnyét is. Ha szeretnénk megváltoztatni egy elem színét, amely a weblapunk több oldalán is megtalálható, akkor azt az összes oldalon át kellene írunk, de ha egy CSS fájlban tároljuk az összes tulajdonságot, akkor csak egy helyen kell változtatnunk a kódon és ezáltal az oldal összes eleme megkapja az adott tulajdonságot. Nem csak a

programozó munkáját könnyíti meg ez az eljárás, hanem az oldal letöltésének ideje is jelentősen csökken, hiszen egy adott CSS fájlt csak egyszer kell letölteni, utána beke-
rül a gyorsítótárba és ha a következő oldalletöltésnél is erre a stíluslapra van szüksé-
günk, nem kell várnunk arra, hogy ismét letöltődjön, hiszen a böngésző gyorsítótára
már tartalmazza azt.

A HTML és a CSS szintaktikája jelentősen eltér egymástól. A stílusokat kijelölők (*selector*) és meghatározások (*declaration*) segítségével lehet megadni.

A kijelölők csoportosítása:

- *Elem kiválasztás*: a legegyszerűbb módszer, mellyel közvetlenül a HTML taget je-
lölhetjük ki és adhatunk meg neki meghatározásokat.

```
body
{
  background-color: #333333;
  font-size: 11pt;
}
```

Ebben az esetben a <body> elemnek adtunk egy szürke háttérszínt és 11-es betű-
méretet.

- *Osztály alapú kiválasztás*: lehetőségünk van arra, hogy az egyes HTML elemeket
osztályokba soroljuk. Ezt úgy tudjuk megtenni, hogy az adott elemnek megadjuk a
class tulajdonságát és hozzá egy értéket, mellyel az adott tulajdonságú elemekre hi-
vatkozunk majd.

```
.felkover
{
  font-weight: bold;
}
```

Ebben a példában a *felkover* nevű osztályokat félkövérré alakítottuk. Az osz-
tálykijelölőket egy oldalon belül több helyen is használhatjuk, és ez még tovább
növeli a kódolás rugalmasságát.

- *Azonosító alapú kiválasztás:* ez a fajta kiválasztó nagyon hasonlít az osztály alapú kiválasztáshoz, mindössze annyi a különbség közöttük, hogy az azonosító kijelölőket egy HTML dokumentumon belül csak egyszer lehet használni. Ezt a kiválasztó egy HTML elem *id* tulajdonságával hozható létre, míg CSS-ben a következőképpen néz ki:

```
#tartalom
{
  width: 560px;
  text-align: justify;
}
```

A `tartalom` elnevezésű azonosítónak megadunk egy 560 pixel szélességet és a szöveg elrendezése sorkizárt lesz.

- *Helyzetérzékeny kiválasztás:* ebben az esetben, a fastruktúrában egy adott elem alatt lévő elemhez rendelhetünk stílust.

```
table a
{
  text-decoration: none;
  color: #000000;
}
```

Ez a példa mutatja a helyzetérzékeny kiválasztást. Jelentése, hogy az oldal táblázataiban lévő összes hivatkozásnál az aláhúzás nem látszik, illetve a szövegszint fekértére állítjuk be. Az oldalon található többi hivatkozásnál ez a stílus nem érvényesül.

- *Szülő-gyermek kiválasztás:* abban különbözik a helyzetérzékeny kiválasztástól, hogy az elemeknek a fában betöltött helye szerinti közvetlen szülő-gyermek kapcsolata esetén tudjuk megadni a stílusmeghatározását.

```
td>a
{
  text-decoration: none;
  color: #000000;
}
```

Amennyiben az előző példánál maradunk, akkor itt közvetlenül a <td> elemekben található hivatkozásokat alakítjuk át a meghatározottak szerint. Ezt a lehetőséget a CSS 2-es, vagy újabb verziói támogatják.

- *Tulajdonság alapú kiválasztás:* van még egy olyan lehetőségünk is, amikor egy adott elem valamelyik paraméterének meglétekor, vagy annak egy értéke szerint szeretnénk a tulajdonságát megváltoztatni.

```
input[type="submit"]
{
    margin: 15px 5px 0 0;
    text-align: right;
}
```

A példánkban az elküldő gomb margójának értékeit állítottuk be, illetve jobbra igazítottuk az oldalon. Szintén a CSS 2-es verziójától működnek ezek a kijelölők.

Ahogy láthattuk a CSS hatalmas segítséget jelent egy oldal megfelelő megjelenésének előállítására. Nem csak a megjelenést, hanem az oldal struktúráját is könnyedén kódolhatjuk segítségével. A mostanában divatos honlapok általában egyfajta elrendezést követnek. Ez a fajta elrendezés a *10. ábrán* látható. Ha egy weblap igazán ki akar tűnni a többi közül, akkor egy olyan design-ra van szükség, ami felkelti az emberek figyelmét. Ez lehet a színek nem hagyományos alkalmazása, de nagyon fontos, hogy ekkor is ügyelni kell a harmóniára. Egy másik lehetőség a különleges elrendezés. Sokkal inkább teret hódítanak a leegyszerűsített oldalak, így a látogató teljesen a tartalomra koncentrálhat, ahogyan azt a *11. ábra* is mutatja. A saját webshop készítésekor is megpróbáltam a hagyományostól eltérő elrendezést kialakítani és egy letisztult, könnyen átlátható oldalt kialakítani. A kezdőlap kinézete megtekinthető a *12. ábrán*.

4.3. PHP, Apache és MySQL

A fejlesztés megkezdése előtt nem árt telepíteni egy megfelelő környezetet, amellyel a saját gépünkön keresztül tudjuk futtatni a php fájljainkat. Ezáltal nincs szükség rá, hogy minden módosítás alkalmával feltöltsük tárhelyünkre a dokumentumokat, hanem egy szimpla frissítéssel azonnal megtekinthetjük az eredményt. Az interneten többféle szabadon hozzáférhető környezet letölthető, én az XAMPP 1.7.2-es

verzióját használtam, amely szintén szabad licence-el rendelkezik és letölthető Linux, Windows, Mac OS X és Solaris operációs rendszerekhez. Tartalmazza a PHP 5-öt, az Apache 2.2.14-es webszervert, a MySQL 5.1.41-es verzióját, a phpMyAdmint, valamint egyéb segédprogramokat, amelyekre a fejlesztés során nem volt szükségem. Az elkészült oldal az ATW ingyenes tárhelyére lett feltöltve fel.

A *PHP* egy szerver oldali szkript nyelv, amit jellemzően a HTML oldalakkal szoktak kombinálni. Manapság az egyik legelterjedtebb programnyelv, amellyel weblapokat készítenek. A PHP-n kívül használják még a Perl-t, a Python-t, illetve a Visual Basic-et is. Mindegyik nyelv könnyen tanulható, de számomra a legjobb megoldásnak mégis a PHP bizonyult. Az interneten rengeteg referencia, és leírás található hozzá. A legkülönbélebb megoldások is részletesen és nem utolsó sorban magyar nyelven fellelhetőek minimális keresgélés után. Jelenleg a PHP 5 a legújabb verzió, amely jelentősen gyorsabb lett az előzőekhez képest, illetve sokkal nagyobb támogatást kapott az objektum orientált szemléletmód is. Rengeteg beépített függvénnyel rendelkezik, melyek segítik a webszerver konfigurációját, illetve az adatbázisok kezelését is.

A PHP szintaktikája viszonylag egyszerűnek mondható, sokban hasonlít a Visual Basic-hez.

Egy egyszerű Hello World kiíratás:

```
<?php
    $uzenet = "Hello World!";
    echo $uzenet;
?>
```

Minden PHP szkriptet a `<?php ... ?>` jelek határolnak és minden utasítás végét pontosvesszővel zárjuk le. Az egyes utasításblokkokat kapcsos zárójelek közé kell helyezni.

Megjegyzéseket `//` jellel szűrhatunk be, valamint a `/* */` jelek közé illesztve is megoldható.

Változókat a `$` jel használatával hozhatunk létre, ahogyan a példában is látható `$uzenet` változó.

A PHP-ben a változók típusát nem szükséges meghatározni. A szöveges változók értékét mindig idézőjelek közé tesszük, míg a numerikus típusoknál nincs szükség idézőjel használatára.

Mivel alkalmazásunkat PHP-ben írjuk, és egy böngészőben szeretnénk futtatni a létrehozott szkriptjeinket, ezért szükségünk van egy webserverre, hiszen a böngészőnk egy web kiszolgálóval szeretne kommunikálni. Erre a legegyszerűbb és egyben a legjobb megoldást az Apache szerver biztosítja, mert a legtöbb ingyenes és fizetős tárhelyen is Apache-ot működtetnek.

Az Apache HTTP Server (röviden Apache) egy nyílt forráskódú webkiszolgáló alkalmazás, szabad szoftver, mely kulcsfontosságú szerepet játszott a World Wide Web elterjedésében. A projekt célja egy olyan webserverver program létrehozása, karbantartása, és fejlesztése, amely megfelel a gyorsan változó Internet követelményeinek, biztonságos, üzleti, vállalati felhasználásra is megfelelő és szabadon használható. Az Apache a régi NCSA httpd szerverre épül, az Apache Szoftver Licenc feltételei alatt terjesztik.

Apache-on kívül használhatunk még egyéb elterjedt webkiszolgálókat is, mint például az Internet Information Services (IIS), amely a Microsoft-os rendszerekben található meg, az Nginx, a Lighttpd, a Zeus Web Server, a Sun ONE, vagy a szabad szoftverek közül a Cherokee HTTP Server, vagy a Xerver.

Egy dinamikus weboldal létrehozásához elengedhetetlen az adatbázisok alkalmazása. A jelenleg legelterjedtebb megoldás erre a MySQL alkalmazása. A MySQL egy többfelhasználós, többszálú, SQL-alapú relációs adatbázis-kezelő szerver.

A MySQL-en kívül egy másik nagyon népszerű adatbázis-kezelő rendszer a PostgreSQL, mely egy szinte minden platformon működő nagyteljesítményű, megbízható és stabil alkalmazás. Ezen kívül még használatos a Microsoft-os környezetekben az MS-SQL. Azért döntöttem mégis a MySQL mellett, mert nagyobb tapasztalattal rendelkezem a használatában és a legtöbb tárhelyen is ezt a rendszert használják. Egyszerű és könnyen kezelhető felületet biztosít hozzá a phpMyAdmin, amely szintén nyílt forrású és rengeteg hasznos funkcióval könnyíti meg az adatbázisok használatát.

4.4. Keresőoptimalizálás

A keresőoptimalizálás, vagy más néven SEO (*Search Engine Optimization*) manapság már egy elengedhetetlen eszköz, ha szeretnénk, hogy a webes tartalmainkat mások is megtalálják, és oldalainknak látogatói legyenek. A SEO maga tulajdonképpen csak egy szelete annak a hatalmas tortának, amely arra szolgál, hogy olyan weboldalt hozunk létre, amelyek a felhasználók által könnyen megtalálhatóak.

A könnyen megtalálható webhely kialakítása az alábbi részterületeket érinti:

- Információs architektúrák
- Fejlesztés
- Piackutatás (marketing)
- Szövegírás
- Látványtervezés
- Keresőoptimalizálás (Search engine optimization; SEO)
- Hozzáférhetőség
- Használhatóság

Ha szeretnénk, hogy oldalunk minél látogatottabb legyen, akkor a legnagyobb felelősség a fejlesztőket terheli. Éppen ezért csak a fejlesztőkre vonatkozó optimalizálási feladatokat szeretném nagyvonalakban bemutatni, amiket a webáruház kódolásánál is figyelembe vettem.

Egy SEO szempontból tökéletesen megírt kód már elegendő, hogy jó helyet érjünk el a keresők találati listájában. Az, hogy mit értünk „jó hely” alatt már kicsit kérdésesebb. Statisztikai adatok alapján a keresőt használó emberek nagy része, csak az első oldalon lévő találatokat tekintik relevánsnak. Tehát aki látogatókat szeretne szerezni, annak az első 20, de inkább az első 10 találatban benne kell lennie. A 13. ábrán látható az első 10 helyre való kattintások eloszlása százalékban. Ebből egyértelműen leszűrhetjük, hogy az első 3 hely a legértékesebb. A lista első oldala a találatok több, mint 91 százalékát fedi le. Éppen ezért, ha optimalizálni szeretnénk weboldalun-

kat, akkor először is tudnunk kell, hogy a keresőrobotok pontosan mit is keresnek egy oldalon, mi az, ami előre lendíthet minket néhány találattal, mire kell odafigyelnünk a fejlesztés során?

A keresőoptimalizáció két nagy részből tevődik össze. Az onsite és az offsite optimalizálásból.

4.4.1. Onsite optimalizáció

Az onsite optimalizálás, azaz az oldalon belüli optimalizálás, a weboldal átalakítása, tartalmi és szerkezeti módosítása annak érdekében, hogy minél relevánsabb legyen az oldal az adott kulcsszóra.

Akkor érhetünk el nagy sikereket, ha minél többször szerepeltetjük a kulcsszavakat és minél kiemeltebb helyeken:

- *Domain*: nagyon fontos, hogy megfelelő domain-nevet válasszunk, mert a legjobb, ha a kulcsszavunk szerepel benne. Sajnos az Ital Futárnál a legjobb nevek már le voltak foglalva, ezért ezt a módszert nem tudtam alkalmazni. Azonban lehetőség van arra, hogy megvásároljunk mások által már nem használt domain-eket. Ehhez viszont valószínűleg sok pénzre lesz szükségünk, mert a gyakori keresőszavakat tartalmazó neveket milliókért árulják.
- *URL*: ha nem sikerült olyan domain-t regisztrálni, amely tartalmazná a legfontosabb kulcsszavunkat, akkor érdemes legalább az URL-ben szerepeltetni. Ezt a módszert hívják keresőbarát linkeknek. Ehhez vagy a fájlneveinket választjuk meg úgy, hogy azok kulcsszavakat tartalmaznak, vagy a webservert .htaccess fájlját alkalmazzuk. Ennek a fájlnak több felhasználási lehetősége is van, mint például egy adott oldal átirányítása, vagy a 404-es hibaoldal egyéni kialakítása. Sajnos ezt a lehetőséget sem tudtam kihasználni, ugyanis az ATW ingyenes tárhelye nem engedélyezte a .htaccess fájl futtatását a szerveren.
- *Title tag*: ebben a tag-ben adhatjuk meg az oldal nevét, amely a böngésző fejlécében, valamint a könyvjelzők nevéként jelenik meg. Létrehozhatunk úgynevezett dinamikus címeket is, melyek főként az egyes termékoldalaknál hasznosak, ugyanis az oldal címébe belefoglalja a termék nevét is, így egy speciáli-

sabb keresés alkalmával nagyobb az esély, hogy a mi oldalunkat fogja jobb helyre rakni a kereső. A legtöbb ingyenes tárhelyen, mint például az ATW, ez sem megoldható, mert az összes weblapunknak egy címe lesz a keretes oldal-szerkezet kialakítása miatt.

- *Header tag*: ezek közül is a legfontosabban a <h1> és <h2> tag-ek, ugyanis ezek az egyes lapok címét jelentik. Nagyon fontos, hogy a <h1> tag-ből egy lapon csak egyet használjunk, mert a keresőrobotok nem tolerálják a többszöri alkalmazását, hiszen egy lapnak csak egy címe lehet. A <h2> tag-ből már alkalmazhatunk többet is az alcímek kialakításához.
- *Meta-tag*: az egyes meta adatok ma már nem olyan nagy jelentőségűek a keresőrobotok számára, mint régebben, ugyanis könnyen manipulálhatóak. Ennek ellenére a *meta description* tag még mindig használatos, mert a keresőlista megjelenítésekor, az ebben a tag-ben található leírás jelenik meg az oldal címe alatt.
- *Linkek*: a linkek nevében is célszerű kulcsszavakat szerepeltetni, mert ezzel is hozzájárulhatunk a minél jobb helyezés eléréséhez.
- *Képek*: a képek *alt* tagját nem csak azért érdemes használni, mert ha egy kép nem töltődik be, akkor az ebben szereplő szöveget jeleníti meg az oldal, hanem a robotok számára is fontos információkat jelent az oldalak indexelésékor.

4.4.2. Offsite optimalizáció

Az offsite optimalizáció nem más, mint a weboldalakon kívül történő optimalizálás. Ugyanis minél több és minél jobb minőségű link mutat az oldalunkra, annál előrébb kerülhetünk a találati listában. Ezzel a lehetőséggel foglalkozik a linkmarketing, mely az oldalra mutató linkek szerzéséről szól. Ez egy olyan folyamat, ami egy weboldal egész életét végigkíséri, ugyanis a linkekből soha nem elég, főleg ha az adott területen nagy a verseny. Az, hogy egy oldal milyen minőségű, azt az úgynevezett „*pagerank*” határozza meg. A pagerank a Google értékelése 0-tól 10-ig. Minél előrébb vagyunk a keresőkben, minél több és minél jobb minőségű oldalak linkelik a mi oldalunkat, annál nagyobb a pagerank értéke. Ezekon kívül még sokfé-

le paramétere van az értéknek, de néhányat nem hoz nyilvánosságra a Google, csak következtetni lehet rájuk. Egy 6-os pagerank-ú oldal már magyar szinten elég erősnek számít.

Látható tehát, hogy a keresőoptimalizálás egy külön szakmának számít manapság, de egy weboldal életében elengedhetetlen kelléke annak, hogy a lehető legtöbb látogató nézzon be a webboltunkba és ezekből a látogatókból minél többet vásárlásra csábítsunk.

5. A webáruház megvalósítása

A webáruház kódolása viszonylag sok időt vett igénybe, hiszen csak nagyon minimális programozási tapasztalattal rendelkezttem a munka megkezdése előtt. Az alkalmazás írásakor több problémába is ütköztem, aminek megoldása nem mindig sikerült, de igyekeztem a lehető legjobb tudásom szerint elkészíteni az oldalt.

5.1. Az adatbázis tervezése és elkészítése

A legfontosabb ok, amiért adatbázis-tervezéssel kell foglalkoznunk, az, hogy csak így biztosíthatjuk az adatbázisban tárolt adatok következetességét, épségét és pontosságát. A pontatlan adatok minden bizonnyal a rossz adatbázis-tervezés legártalmasabb következményei. Éppen ezért lépésről lépésre szeretném bemutatni, hogyan alakítottam ki a webáruház adatbázisát.

5.1.1. Előzetes táblaszerkezet meghatározása

Mielőtt hozzákezdenék bármilyen számítógépes alkalmazás elindításához, szerintem a legjobb módszer, ha papíron megtervezzük az adatbázis szerkezeti felépítését nagyvonalakban. Éppen ezért elgondolkoztam, hogy milyen táblákra is lehet majd szükségem és ezeket leírtam. Minden egyes tábla alá leírtam a hozzá tartozó mezőnevek listáját is. Ez a lista a későbbiekben még sokat bővült, de egy kiindulási alapnak teljesen megfelelt.

Az Ital Futár előzetes táblaszerkezete mezőnevekkel

Ügyfelek	Adminok	Termékek	Rendelések	Kosár
Ügyfél neve	Admin neve	Termék neve	Ügyfél neve	Termék neve
Ügyfél email	Admin email	Termék leírása	Termék neve	Termék ára
Ügyfél lakcíme		Termék ára	Fizetendő	Mennyiség
Ügyfél telszáma		Kategória	Rendelés napja	
Regisztráció				

5.1.2. A normalizálás folyamata

A normalizálás az adatbázisban található adatok rendszerezését jelenti. Bizonyos előre meghatározott szabályok szerint megszüntetjük a redundanciát és az inkonzisztenciát az adatbázisban. Redundanciáról akkor beszélünk, ha valamilyen adatot többször is eltárolunk. Ennek megszüntetése azért fontos, mert egy redundáns adattáblában az adatok törlése, módosítása, frissítése meglehetősen nehézkes, valamint a tárolónk kapacitását is feleslegesen foglalják az ismételt adatok. Egy adatbázist akkor nevezünk inkonzisztensnek, ha egymásnak ellentmondó tényeket tartalmaz.

Első normálforma: egy reláció első normálformában van, ha

- az egymáshoz kapcsolódó attribútumok külön táblával rendelkeznek,
- az egyes táblákban nincsenek ismétlődő attribútumok,
- a táblák rendelkeznek elsődleges kulccsal.

Második normálforma: egy reláció második normálformában van, ha

- első normálformában van,
- a több rekordra vonatkozó attribútumok külön táblákban vannak.

Harmadik normálforma: egy reláció harmadik normálformában van, ha

- második normálformában van,
- nincsenek olyan attribútumok, amelyek nem függenek a kulcstól.

Az előzetes táblaszerkezet meglehetősen sok redundáns adatot tartalmazott, ezért a tervezés során ezeket a redundanciákat folyamatosan próbáltam kiküszöbölni és törekedtem rá, hogy a tábla szerkezete minél inkább megfeleljen a harmadik normálformának.

5.1.3. A végleges adatbázis felépítése

A végleges adatbázis már több mezővel rendelkezik, ugyanis az egyes funkciók, mint például a beléptetés, vagy a regisztráció miatt újabb attribútumokat kellett beírnom a táblákba.

Az **ÜGYFELEK** tábla: azoknak az embereknek az adatait tartalmazza, akik regisztráltak az oldalon. A tábla adatainak segítségével kapcsolatba tudunk lépni az adott személlyel és elküldhetjük neki a legjobb ajánlatainkat.

Attribútum neve	Attribútum típusa	Attribútum leírása
id	int (10)	Az ügyfél azonosítója.
veznev	varchar (100)	Az ügyfél vezetékneve.
kernev	varchar (100)	Az ügyfél keresztnéve.
email	varchar (50)	Az ügyfél email címe.
jelszo	varchar (100)	Az ügyfél jelszava.
session_id	varchar (150)	Az ügyfél munkamenet azonosítója.
reg_idopont	datetime	Regisztráció időpontja.
belepve	datetime	Az utolsó belépés időpontja.
ervenyes	datetime	A munkamenet azonosító érvényességi időpontja.
kilepve	datetime	Utolsó kilépés időpontja.
vez_tel	varchar (30)	Vezetékes telefonszáma.
mobil_tel	varchar (30)	Mobiltelefon száma.
orszag	varchar (100)	Az ügyfél szállítási címe. – ország
irszam	varchar (4)	Az ügyfél szállítási címe. – irányítószám
varos	varchar (100)	Az ügyfél szállítási címe. – város
utca	varchar (50)	Az ügyfél szállítási címe. – utca
hszam	varchar (10)	Az ügyfél szállítási címe. – házszám
sz_nev	varchar (100)	Az ügyfél számlázási neve.
sz_irszam	varchar (4)	Az ügyfél számlázási címe. – irányítószám
sz_varos	varchar (100)	Az ügyfél számlázási címe. – város
sz_utca	varchar (50)	Az ügyfél számlázási címe. – utca
sz_hszam	varchar (10)	Az ügyfél számlázási címe. – házszám

Az **ADMINOK** tábla: azoknak a felhasználóknak az adatait tartalmazza, akik be tudnak lépni az adminisztrációs felületre és ott módosíthatják a termékek paramétereit, valamint menedzselik a rendelés folyamatát.

Attribútum neve	Attribútum típusa	Attribútum leírása
id	int (10)	Az adminisztrátor azonosítója.
email	varchar (50)	Az adminisztrátor email címe.
password	varchar (30)	Az adminisztrátor jelszava.

A **TERMÉKEK** tábla: ebben a táblában találhatóak maguk az italok és azok paramétereit. Ezáltal könnyen kilistázhatjuk a webáruházban az italokat és minden más velük kapcsolatos adatot.

Attribútum neve	Attribútum típusa	Attribútum leírása
id	int (10)	A termék azonosítója.
nev	varchar (50)	A termék neve.
kateg	int (2)	A termék kategóriájának száma.
kis_kep	varchar (200)	A termék kis képének az url címe.
nagy_kep	varchar (200)	A termék nagy képének az url címe.
rovid_leiras	varchar (300)	A termék rövid leírása.
hosszu_leiras	longtext	A termék hosszabb leírása.
ar	int (10)	A termék ára.
min_egyseg	int (2)	A termék minimum rendelhető mennyisége.
egyseg	varchar (10)	A termék rendelhető mértékegysége.
raktaron	tinyint (1)	Megjelölhetjük, hogy van-e raktáron az adott termékből.
felvitel_date	datetime	A termék felvételének dátuma.

A **RENDELÉSEK** tábla: ez a tábla a rendelések nyilvántartására szolgál. Azért van rá szükség, hogy a termék kiszállítása és átvétele zökkenőmentesen történjen.

Attribútum neve	Attribútum típusa	Attribútum leírása
id	int (10)	A rendelések azonosítója.
ugyfel_id	int (10)	A megrendelő ügyfél azonosítója.
fizetendo	int (10)	Fizetendő végösszeg.
code	bigint (20)	Egy véletlen számot tárolunk benne, amellyel megakadályozható, hogy a rendelést többször elküldje a vásárló.
rendeles_date	datetime	A rendelés dátuma.

A **KOSÁR** tábla: az ügyfelek által kosárba helyezett termékeket tartalmazza. Minden egyes ügyfél saját kosárral rendelkezik, és a meg nem vásárolt, de kosárba helyezett termékek az oldal ismételt meglátogatásával, ugyanúgy megjelenik a kosárban.

Attribútum neve	Attribútum típusa	Attribútum leírása
id	int (10)	A kosár azonosítója.
termek_id	int (10)	A kosárban lévő termék azonosítója.
ugyfel_id	int (10)	Annak az ügyfélnek az azonosítója, aki a kosárba helyezte a terméket.
rendeles_id	int (10)	Megjelöljük, hogy az éppen kosárban lévő termék már meg van-e rendelve.
session_id	varchar (150)	Az aktuális munkamenet azonosító.
mennyiseg	int (5)	A kosárba helyezett termék mennyisége.
kosarba_date	datetime	A kosárba helyezés dátuma.

Ezzel az adatbázis meg lett tervezve, mindössze annyi dolgunk maradt vele, hogy fizikailag is megvalósítsuk. Ehhez a phpMyAdmin-t használtam, mellyel egy grafikus felhasználói felületen tudtam implementálni az adatbázisomat.

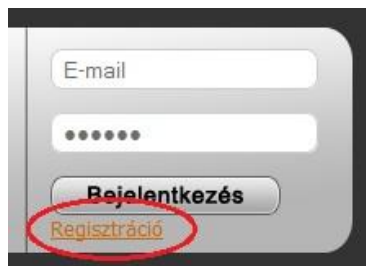
5.2. Felhasználói felület létrehozása

Az egész webáruházat úgy terveztem meg, hogy elkülönüljön a felhasználói és az adminisztrációs felület. Mind a két felület kialakításánál figyelembe vettem, hogy

minél kényelmesebb legyen az oldal használata és a funkciók könnyen elérhetőek legyenek. Az oldal grafikus elemeit az Adobe Photoshop szoftverrel hoztam létre.

5.2.1. Regisztráció és belépés

Ahhoz, hogy megrendelhesünk egy terméket az oldalon, szükséges, hogy regisztrálva legyen az ügyfél az adatbázisban. A regisztráció egy linkre való kattintással kezdhető el, ahogy az a képen is látszik:



A regisztráció

A linkre kattintva, megjelenik egy egyszerű űrlap, amelyben található néhány kötelezően kitöltendő mező. Mind szerver oldalon, mind kliens oldalon odafigyeltem, hogy a regisztráló személy ne adhasson meg helytelen adatokat.

Kliens oldalon a `reg.js` nevű fájlban található egy `ellenoriz()` nevű függvény, amely a Regisztráció gombra való kattintáskor hívódik meg, és addig nem engedi elküldeni a szervernek az adatokat, amíg minden kötelező mező nincs kitöltve, vagy az email címnek helytelen formátum van megadva.

Kódrészlet a `reg.js` nevű fájlból:

```
function ellenoriz()
{
    var mehet = 1;
    if (mehet == 1 && isEmpty(document.urlap.veznev.value))
    {
        mehet = 0;
        document.urlap.veznev.focus();
        alert('Kérlek add meg a vezetéknévedet!');
    }
}
```

```

[...]
if (mehet == 1)
{
    document.urlap.submit();
}
else
{
    document.urlap.elkuldgomb.value='Regisztráció!';
    document.urlap.elkuldgomb.disabled=false;
}
}

```

A függvény elején a mehet nevű változónak 1 értéket adunk, és ha az űrlap bármely eleménél valamilyen hibát észlelünk, akkor 0-ra állítjuk. A függvény végén, ha a mehet értéke 1, akkor az űrlapot elküldjük a szervernek.

A szervertől oldali ellenőrzés a JavaScript fájl helyes lefutása után kezdődik, ugyanis az űrlapot átirányítjuk a `reg_ellenoriz.php` fájlra. Nagyon fontos, hogy ne csak kliens oldali ellenőrzést végezzünk, hiszen minden böngészőben ki-kapcsolható a JavaScript futtatása, és ebben az esetben az ellenőrzés nem történik meg, így akármilyen adatokat elküldhetünk a szervernek, amellyel komoly károkat okozhatnak az adatbázisunkban az esetleges rosszindulatú felhasználók. Ebben az ellenőrzési fázisban először megvizsgáljuk, hogy megkaptuk-e az adatokat az űrlapból. Amennyiben nem kaptunk semmilyen adatot, abban az esetben visszairányítjuk a felhasználót a nyitó oldalra és befejezzük az oldal futtatását. Ha kaptunk adatokat, akkor lekérdezzük az adatbázis, `ugyfelek` nevű tábláját, és megvizsgáljuk, hogy a kapott email cím szerepel-e benne. Ha valaki regisztrált már ezzel a címmel, akkor egy egyszerű üzenettel a felhasználó tudomására hozzuk ezt, viszont ha még nem regisztráltak ezzel a címmel, akkor a bevitt adatok megfelelő vizsgálata után, beírjuk az `ugyfelek` táblába az adott sort, majd átirányítjuk a kezdőoldalra a felhasználót. Fontosnak tartom a felhasználók jelszavainak biztonságos tárolását, ezért a PHP `md5()` függvényével titkosítottam a jelszavakat. Ez a függvény az *md5* titkosítást alkalmazza. Az MD5 (*Message-Digest algorithm 5*) egy 128 bites, egyirányú kódolási algoritmus. Az RFC 1321-es internet szabványban foglaltak

szerint használják internetes adatok kódolására, illetve titkosítási kulcsokban. Az MD5-kódolást biztonsági alkalmazások széles skálája használja adatellenőrzésre, például fájlok eredetiségének (sértetlenségének) vizsgálatára.

A regisztrációs űrlap jelszavas beviteli mezőjének értékére alkalmazzuk az *md5* titkosítást, majd az így kapott sztringet tároljuk el az adatbázisban a többi adattal együtt. Ezáltal a jelszót már csak nehezen lehet visszafejteni.

Az oldalra való belépéshez mindössze egy regisztrált email címre, valamint egy jelszóra van szükség. A felhasználó hitelesítése itt már csak szerver oldalon történik. A két beviteli mezőben kapott email címet és jelszót megvizsgáljuk, hogy szerepelnek-e az adatbázisban. A beviteli mezőben kapott jelszóra ismét alkalmazzuk az *md5* () függvényt, majd az így kapott sztringet hasonlítjuk össze az adatbázisban eltárolt sztringgel. Ha az email cím és a jelszó is megegyezik, akkor a belevélve változónak 1 értéket adunk, eltároljuk az adatbázisban a belépés idejét és a böngésző munkamenet azonosítóját, valamint sütiben tároljuk el az email címet, és a jelszót, hogy a belépett felhasználót a későbbiekben is megfelelően tudjuk azonosítani. Az azonosítást minden egyes oldal meglátogatásakor elvégezzük.



A bejelentkezett felhasználó üdvözlése

5.2.2. A kosárba helyezés művelete

Az oldalon több helyen is megtalálható a „Kosárba” feliratú gomb. Erre kattintva a vásárló egy átmeneti tárolóba helyezheti a megrendelni kívánt italokat, majd a „Megrendel” gombra kattintva véglegesítheti a vásárlási folyamatot. A „Kosárba” gomb hatására meghívódik a *kosarba.php* fájl. Bárki helyezhet termékeket a kosárba attól függetlenül, hogy be van-e lépve, vagy sem, ugyanis, ha a vásárló nincs bejelentkezve, akkor a session azonosítóját tároljuk el, ha pedig be van jelentkezve, akkor az ügyfél azonosítóját rendeljük hozzá az adott termékhez. Ezáltal

minden egyes oldalfrissítés alkalmával meg tudjuk őrizni a kosár tartalmát. Sőt, ha a bejelentkezett felhasználó a rendelési folyamat elvégzése nélkül jelentkezik ki, legközelebbi bejelentkezése alkalmával, a termékek ugyanúgy megtalálhatóak lesznek a kosarában, ha esetleg be szeretné fejezni a vásárlást.

Lehetőség van arra is, hogy egyes elemeket töröljünk a listából, ehhez mindössze egy piros „x”-re kell kattintani az ital neve mellett.



A kosár tartalma

5.2.3. A rendelés folyamata

A „Megrendel” gombra kattintva elkezdődik a rendelés folyamata, azaz meghívódik a *megerosit.php*, ahol egy oldalon összegezve megtalálható az összes kosárba helyezett termék, és ha akarjuk, még módosíthatunk az italok mennyiségén. Amennyiben nem vagyunk bejelentkezve, egy felugró ablak figyelmeztet minket, hogy addig nem tudjuk megkezdeni a folyamatot, amíg ezt meg nem tesszük.

A tovább gombra kattintva meg kell adnunk a vásárlói adatokat. Ez egy egyszerű form, ami nagyban hasonlít a regisztráció műveletére. Itt is ellenőrzöm a bevitt kliens és szerver oldalon egyaránt, hogy minden adat megfelelően legyen eltárolva az adatbázisban. Ha már vásárolt az oldalon, abban az esetben automatikusan beíródnak az adatok.

A következő oldalon még utoljára ellenőrizheti a vásárló a megrendelését. Ha helyesnek talál mindent, a rendelés elküldésével véglegesíthető a vásárlás.

Fontosnak találtam, hogy egy rendelést többször ne lehessen elküldeni, ezért az összegző oldalon generálok egy véletlen számot, amit eltárolok a *rendelések* tábla *code* mezőjében. Így ha visszalépünk egy oldalt, majd újra megpróbáljuk elküldeni az adatokat, vagy csak frissítjük az utolsó lapot, nem küldi el újra a megrendelést, mert ismét generálódik egy szám, és egy egyszerű szelekciós utasítás visszairányítja a kezdőlapra a felhasználót.

5.3. Adminisztrációs felület létrehozása

Az oldalon található termékek megfelelő menedzselése érdekében fontosnak tartottam, hogy egy viszonylag könnyen kezelhető és az oldaltól teljesen elkülönülő adminisztrációs felületet is létrehozzak. Erre a felületre, csak azok tudnak belépni, akik email címe és jelszava megtalálható az *admin* táblában. Ezen az oldalon lehetőség van a termékek feltöltésére, illetve azok törlésére, valamint ha egy termék éppen nincs raktáron, akkor átmenetileg törölhetjük a webáruházból, viszont az adatbázisban továbbra is megtalálható marad, és bármikor vissza tudjuk állítani.

Ha szeretnénk feltölteni egy adott terméket, akkor a „Termék feltöltése” menüpontra kattintva tehetjük ezt meg. Ez a link meghívja a *termek_feltolt.php* állományt, és az adatok megadása után a „Feltöltés” gombra kattintva, máris elérhetővé válik az adott item a felhasználói felületen. Ezen részfeladat legnehezebb részét a képek feltöltésének megvalósítása okozta, mert egyszerre két képet kellett feltölteni a szerverre.

A képek feltöltésének megvalósítása:

```
$dir = "../images/termekek/$kategoria";

if (check($_FILES['kiskep']) == "rendben" &&
    check($_FILES['nagykep']) == "rendben")
{
    if (move_uploaded_file($_FILES['kiskep']['tmp_name'],
        $dir.$_FILES['kiskep']['name']) &&
        move_uploaded_file($_FILES['nagykep']['tmp_name'],
            $dir.$_FILES['nagykep']['name']))
```

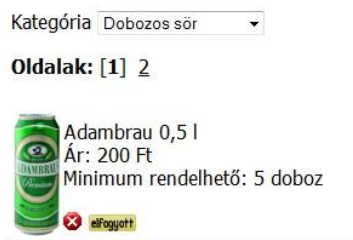
```

    {
        print "<h2>Sikeres feltöltés</h2>";
    }
}
else
{
    print "<h2>Hiba: " . check($_FILES['kiskep'] . "</h2>");
}

```

A képek elérési útvonalát a `$dir` változóban tároltam el. A form elküldése előtt meg kell adnunk, hogy melyik kategóriába sorolandó az adott ital és mivel minden kategóriának van egy saját mappája, ezért a `$kategoria` értéke adja meg az éppen aktuális mappa nevét. A két képet egy asszociatív tömbben tároljuk el és `$_FILES['kiskep']`, valamint a `$_FILES['nagykep']` nevekkel hivatkozhatunk rájuk. A következő lépésben egy szelekcióval megvizsgálom, hogy feltöltésre kerültek-e a képek. Ha a képek megtalálhatóak a szerveren, akkor azokat az átmeneti mappából át kell helyezni a `$dir` által tárolt helyre és a folyamat sikeresen véget ér. Amennyiben nem, akkor a hiba okának kiírásával a futás befejeződik.

A termékek menedzselése menüpontra kattintva törölhetünk egy elemet az adatbázisból vagy csak az Ital Futár oldaláról.



A termékek menedzselése menüpont

Ebben a menüben kategóriánként kilistázódnak az italok és az „x”-re kattintva törölhetőek az adatbázisból, az „elfogyott” gombra kattintva pedig a *termekek* tábla *raktaron* mezőjét állítjuk 0-ra. Mind a két alkalommal egy SQL utasítást adunk ki.

Egy adott termék törlése az adatbázisból:

```
DELETE FROM termekek WHERE id=$id
```

Egy adott termék törlése a webshopból:

```
UPDATE termekek SET raktaron=0 WHERE id=$id
```

Azok az italok is megjelennek az adminisztrációs oldalon, amelyek nincsenek raktáron csak az „elfogyott” gomb helyett egy targonca képe látható, melyre kattintva ismét visszaállíthatjuk az eredeti állapotot.

6. Az oldal tesztelése

Mint minden alkalmazásnál, egy weboldalnál is szükséges a tesztelés, mielőtt használatba helyezzük. A tesztelés folyamatát két részre osztottam, egy design szerinti tesztelésre, valamint a funkciók működésének ellenőrzésére.

Először megvizsgáltam, hogy az elterjedtebb böngészőkben (Internet Explorer, Mozilla Firefox, Google Chrome, Opera, Safari) helyesen jelennek-e meg az oldal elemei, nem-e esik szét az oldal grafikája, ha más felbontású monitorral tekintik meg az oldalt, illetve, hogy a szövegek mindenhol megfelelően olvashatóak. Arra az eredményre jutottam, hogy az Internet Explorer 6-os verzióját kivéve, minden böngészőben helyes a megjelenítés. Mivel a Microsoft megszüntette ennek a verzióknak a támogatását, és a felhasználók mindössze 2-3 százaléka használja már csak ezt a böngészőt, ezért nem is tartottam fontosnak, hogy alternatív CSS fájlokat hozzak létre annak érdekében, hogy az adott verzióban is megtekinthető legyen a weboldal. A 800x600-as felbontású monitoroknál is megfelelően jelentek meg az elemek és a nagyobb felbontásoknál sem tapasztaltam semmilyen elváltozást az oldal szerkezetében. Ezen kívül a megjelenített szövegeket is teljesen olvashatónak és megfelelően kontrasztosnak találtam.

Ezután következett egy próbaregisztráció, valamint egy próbavásárlás és a navigáció tesztelése. Regisztráció során nagy figyelmet fordítottam arra, hogy megvizsgáljam a folyamatot helytelen adatok bevitele esetén is. Az ellenőrzés minden mező esetén megfelelően működött és nem valid adatok esetén, vagy a kötelező adatok megadása nélkül, a regisztráció sikertelen volt. Az oldalra feltöltött termékek hiánytalanul megtalálhatók a megfelelő kategóriákra kattintva. Belépés nélkül megpróbáltam olyan oldalakat elérni, melyek megtekintése regisztrációhoz kötött, de minden próbálkozásomat sikeresen elutasította az alkalmazás és visszairányított a kezdőlapra. Végül tesztelésre került maga a vásárlási folyamat. 10 terméket helyeztem el a kosárba bejelentkezés nélkül, majd bejelentkezés után is, majd kipróbáltam azok eltávolítását és újra a kosárba helyeztem 5 terméket. Ezután következett azok megrendelése. Változtattam a megrendelt mennyiségeken, próbálkoztam helytelen szállítói adatok megadásával, de így nem engedélyezte a megrendelés elküldését, majd helyes adatok megadásával befejeztem a vásárlást. A program tesztelése

során megpróbáltam a lehető legtöbb olyan problémás helyzetet szimulálni, amelyet egy átlagos felhasználó elkövethet, de mindent rendben találtam.

7. Összefoglalás

A szakdolgozatomban az e-kereskedelem informatikai és gazdasági oldalát szerettem volna bemutatni. Gazdasági oldalról megvizsgáltam magának az online kereskedelemnek a kialakulását és legfontosabb előnyeit illetve hátrányait is. Ezáltal olyan következtetésekre jutottam, mely megerősített abban, hogy az interneten való kereskedelemnek még nagy jövője lesz az elkövetkezendő években. Magyarországon az emberek még csak most kezdenek igazán bízni a webáruházak által nyújtott szolgáltatásokban. Nagyon fontos, hogy ezt a lendületet ne törjük meg olyan oldalak létrehozásával, melyek csak a vásárlók pénzét akarják kicsikarni. Bízom benne, hogy egyre több vásárló fogja felismerni a megannyi lehetőséget, amelyet kihasználva, gyorsabban és sorban állás nélkül tudják elintézni a vásárlást és egyéb ügyeiket.

A kérdőív kiértékelése lehetőséget adott arra, hogy megismerjem a magánemberek véleményét az interneten való vásárlásról, saját kialakult szokásaikról, illetve, hogy milyen előnyöket és milyen hátrányokat vélnek felfedezni benne. Ezek alapján próbáltam létrehozni egy olyan webáruházat, amely figyelembe veszi a felhasználók igényeit.

A feladat lehetőséget adott arra, hogy olyan technológiákban és programnyelvekben mélyedjek el, amelyek jelentős szerepet töltenek be a mai informatikai életben. Megismertem, hogy milyen módszerekkel lehet több technikát belevonni egyetlen projektbe. Egyszerre alkalmaztam a HTML, a CSS, a PHP és a MySQL nyújtotta lehetőségeket, ezáltal egy olyan dinamikus oldalt tudtam létrehozni, amelyet némi átalakítással könnyedén lehetne használni a valós életben is. Az elkövetkezendő tanulmányaim során szeretném ezen tudásomat tovább gyarapítani és olyan új ismereteket szerezni, mellyel sokkal több felhasználói igényt lefedő weboldalakat tudok majd létrehozni.

A kódolás során észrevettem, hogy mennyire fontos az egész munkamenet részletes megtervezése. Egy megfelelő terv kialakítása után sokkal gyorsabban és könnyebben befejezhető a feladat. A tervezésben lévő hiányosságok miatt rengeteg időt vett el az előforduló hibák észrevétele és kijavítása.

Sajnos az alkalmazás készítése során a keresőoptimalizálás által nyújtotta lehetőségeket nem tudtam teljes mértékben kihasználni az ingyenes tárhely korlátozásai miatt. Sokkal keresőbarátabb oldalt lehetett volna létrehozni, ha a tárhely nem alkalmazza a keretes oldalszerkezetet és több lehetőséget nyújt a webszerver szolgáltatásainak testreszabásában.

Az oldalt még több funkcióval ki lehetne egészíteni, amely megkönnyíthetné a vásárlást, mint például a kártyás fizetés lehetősége, de mivel egy fiktív cégről van szó, ezt a funkciót nem tudtam implementálni az alkalmazásba.

A későbbiekben tervezem, hogy megvalósítsam az Ital Futár céget a valóságban is, amihez már szükség lesz egy olyan tárhelyre, ahol sikeresen tudom alkalmazni a kimagradt lehetőségeket is. A webshop elérhető a <http://drinkrunner.atw.hu/> weboldalon.

Irodalomjegyzék

Peter Moulding

PHP haladóknak – Fekete könyv

Perfacto-Pro Kft. Budapest, 2002

Virginia DeBolt

HTML és CSS webszerkesztés stílusosan

Kiskapu Kiadó Budapest, 2005

Michael J. Hernandez

Adatbázis-tervezés – A relációs adatbázisok alapjairól földi halandóknak

Kiskapu Kiadó Budapest, 2004

Aarron Walter

Keresők és webhelyek – Webes szabványok és keresőoptimalizálás a gyakorlatban

Kiskapu Kiadó Budapest, 2009

Hámori Balázs, Szabó Katalin

Információgazdaság

Akadémia Kiadó Budapest, 2006

WebShop-Experts Kft.

Webáruház marketing útmutató

(tanulmány)

Dr. Sugár Péter

Elektronikus kereskedelem, EDI, E-bussiness

www.miau.gau.hu/miau/11/sugar.rtf

WebShop-Experts Kft.: Hogyan indítsunk webáruházat?

www.webshopexperts.hu/hogyan_inditsunk_webaruhazat

Weblabor: CSS alapjai I.

<http://weblabor.hu/cikkek/cssalapjai1>

Wikipédia: Apache HTTP Server

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

Wikipédia: MySQL

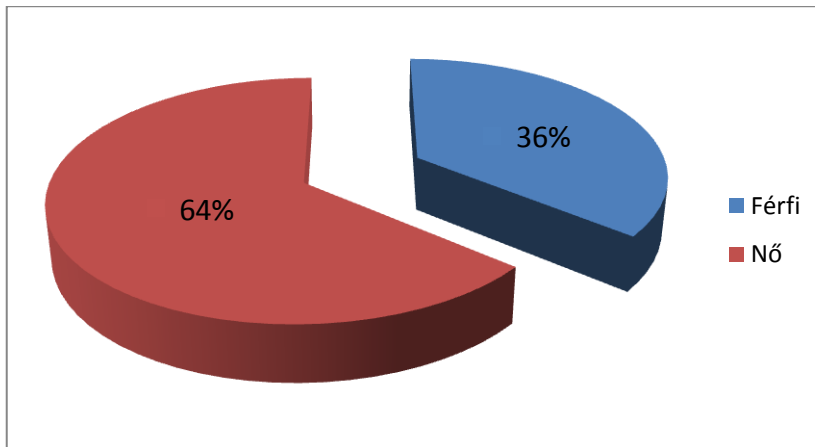
<http://hu.wikipedia.org/wiki/MySQL>

Wikipédia: MD5

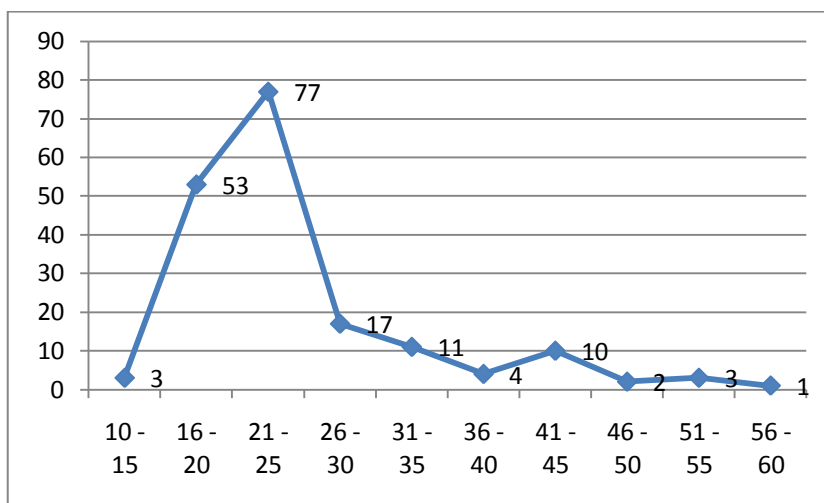
<http://hu.wikipedia.org/wiki/MD5>

Függelék

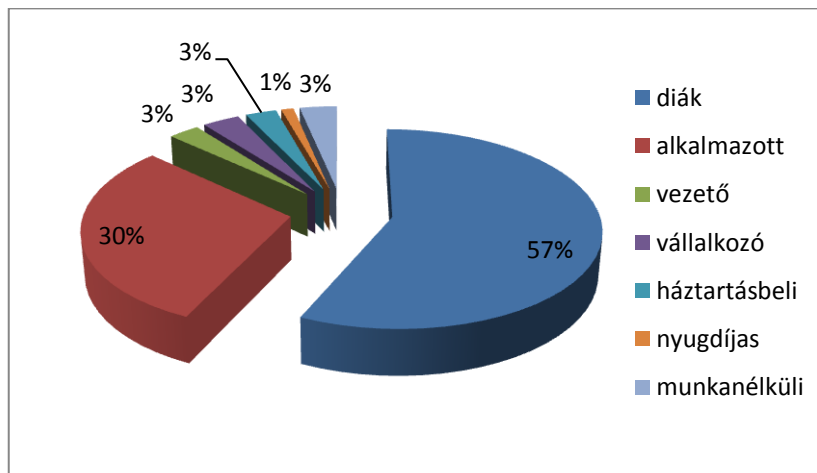
1. ábra: A megkérdezettek nemek szerinti eloszlása



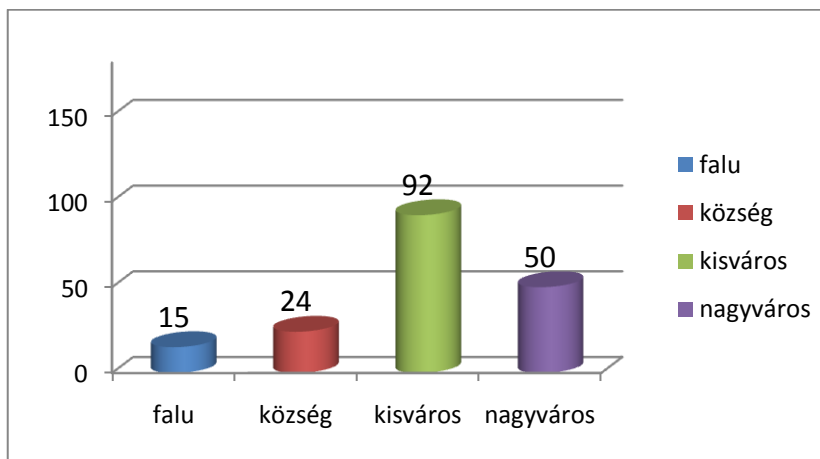
2. ábra: A megkérdezettek korok szerinti eloszlása



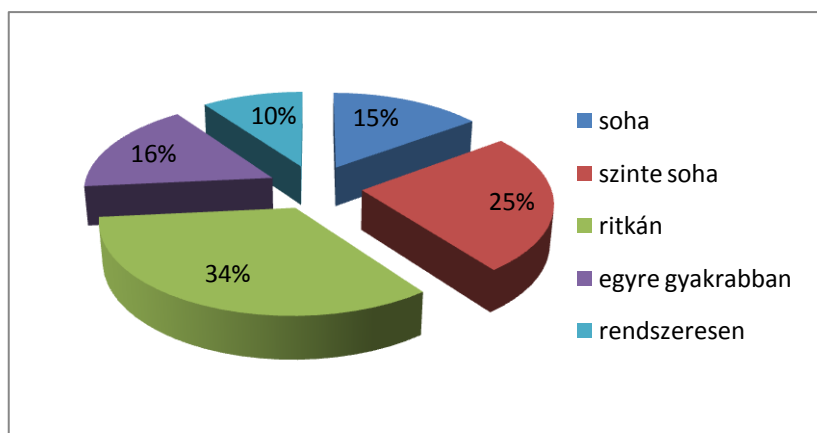
3. ábra: A megkérdezettek foglalkozás szerinti eloszlása



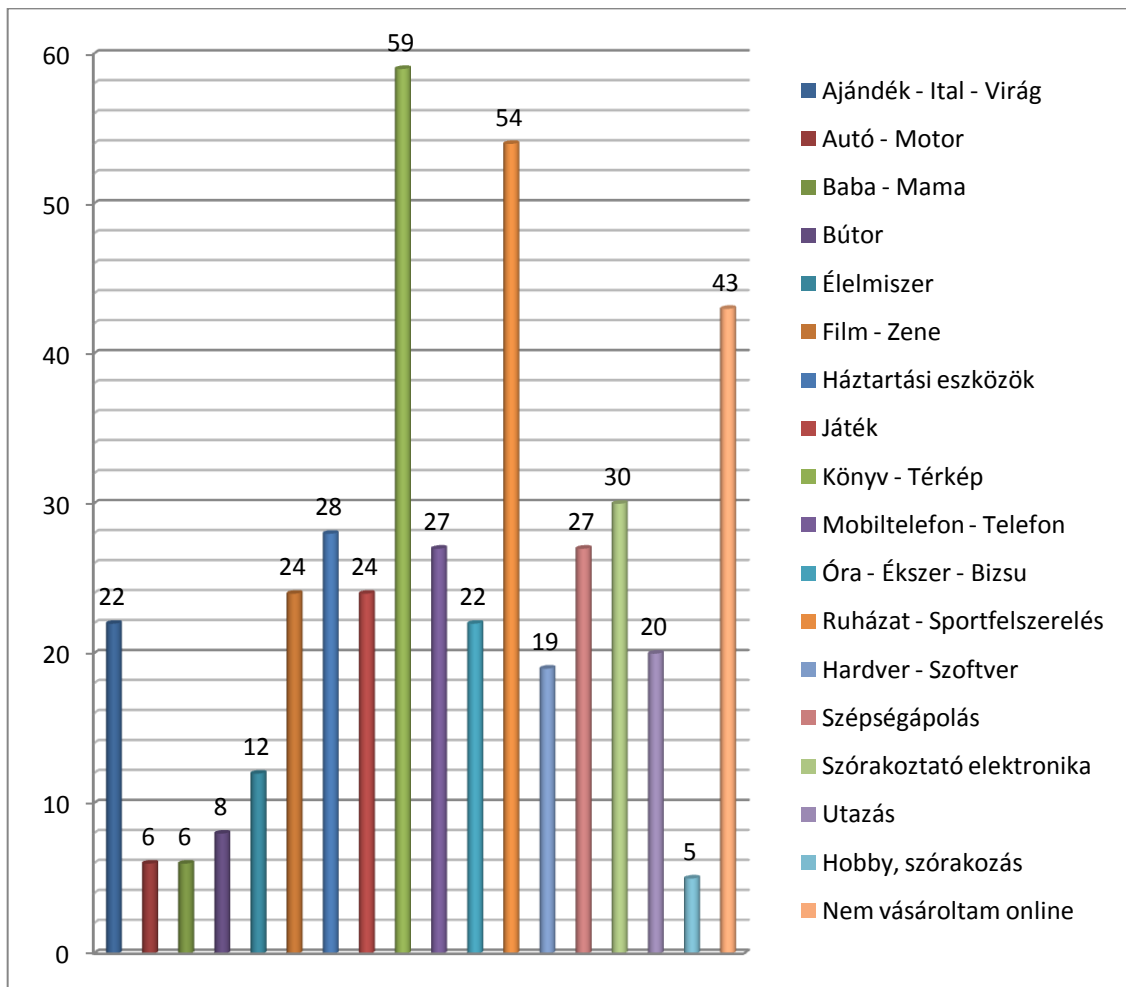
4. ábra: A megkérdezettek lakóhelyük szerinti megoszlása



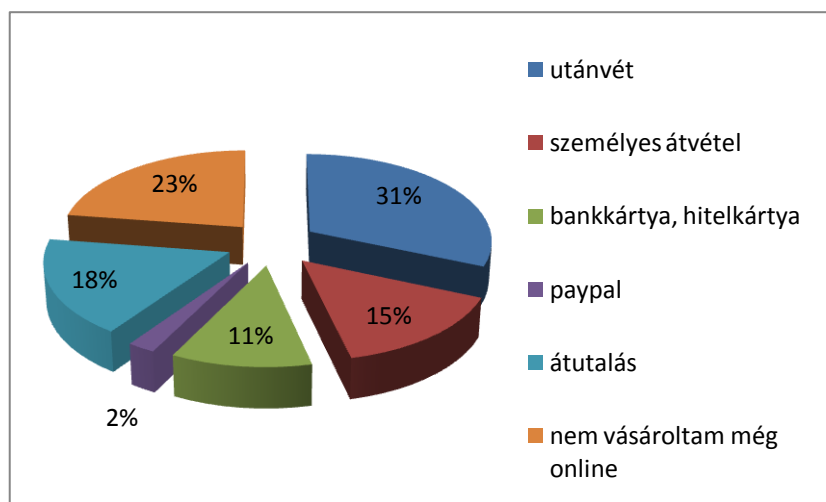
5. ábra: Milyen gyakran vásárolsz az interneten keresztül?



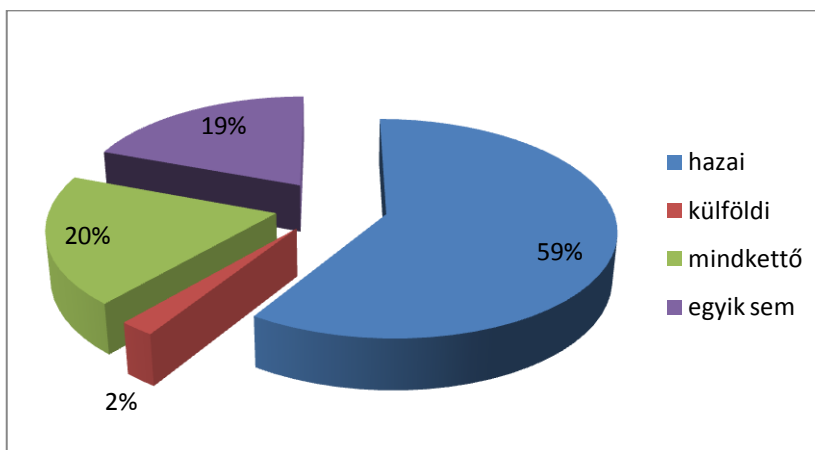
6. ábra: Milyen termékeket vásároltál eddig online áruházakban?



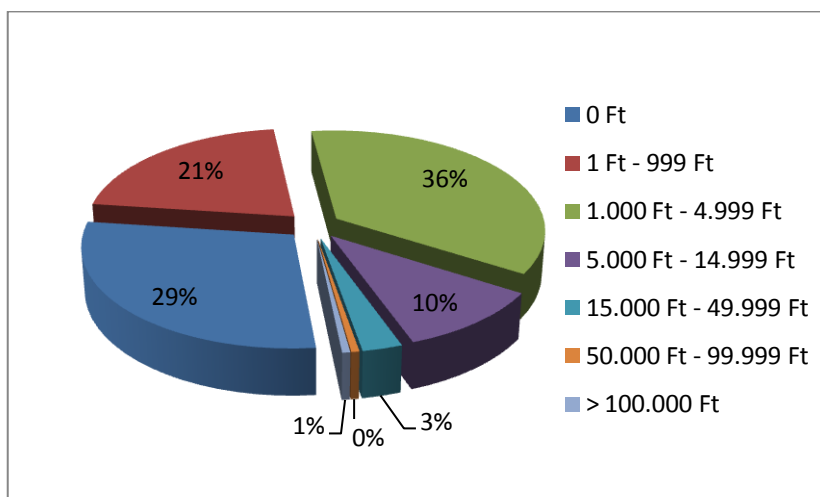
7. ábra: Milyen fizetési módot részesítesz előnyben?



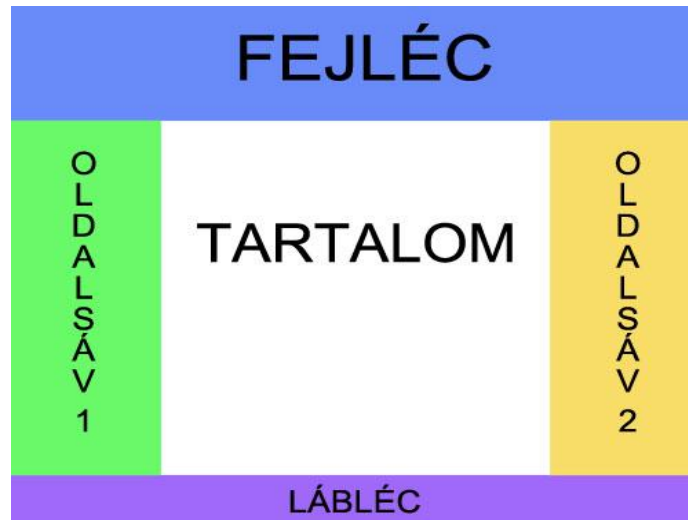
8. ábra: A hazai vagy a külföldi oldalakat részesíted előnyben?



9. ábra: Átlagban mennyi pénzt költesz egy hónapban online vásárlásra?



10. ábra: Oldalterv



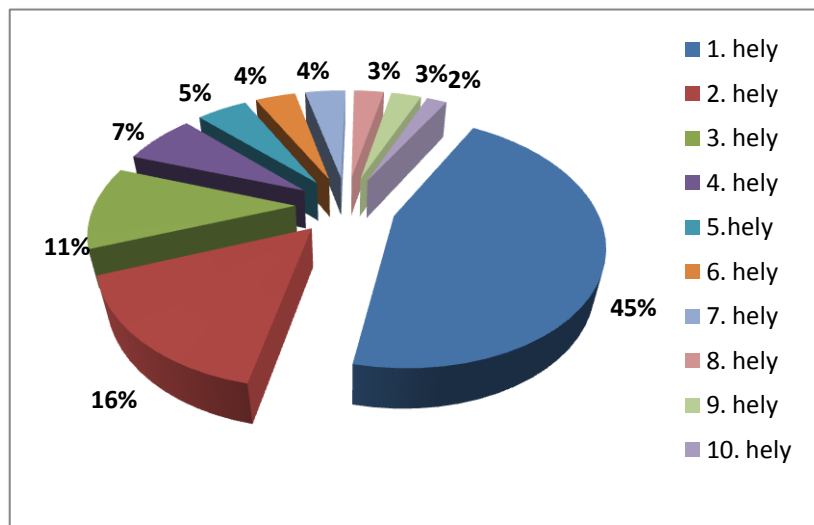
11. ábra: Letisztult weboldalterv



12. ábra: Ital Futár oldalterv



13. ábra: A találati lista hányadik helyére kattintanak?



Köszönetnyilvánítás

Szeretném megköszönni a Mecsei Zoltán témavezetőmnek a dolgozat elkészítéséhez nyújtott segítségét, valamint az összes ismerősömnek, barátomnak és az ő ismerőseiknek, barátaiknak, akik segítségemre voltak a kérdőív kitöltésében.