

SZAKDOLGOZAT

Sárközi Zoltán

Debrecen

2011

Debreceni Egyetem

Informatika Kar

**WEBSHOP ALKALMAZÁS LÉTREHOZÁSA WAMP
KÖRNYEZETBEN**

Témavezető:

Dr. Kuki Attila

Egyetemi adjunktus

Készítette:

Sárközi Zoltán

Mérnök Informatikus

Debrecen

2011

Tartalom

1. Bevezetés.....	5
2. Technológiák ismertetése	7
2.1. HTML.....	7
2.1.1. HTML rövid története	7
2.1.2. A HTML nyelv alapjai	7
2.2. PHP.....	9
2.2.1. A PHP története	9
2.2.2. A PHP nyelv alapjai	10
2.3. MySQL.....	11
2.3.1. A MySQL fejlődése.....	11
2.3.2. A MySQL jellemzői	12
2.4. Mi a CSS?.....	13
2.5. JavaScript	16
3. Felhasznált eszközök	18
3.1. Marcomedia Dreamweaver MX 2004.....	18
3.2. WampServer.....	18
3.3. PhpMyAdmin	19
4. Az adatbázis	20
4.1. Az adatbázis alapfogalmak.....	20
4.2. A táblák létrehozása	21
4.3. A táblák szerkezete.....	22
4.4. A táblák kapcsolata	26

5. Az alkalmazás felépítése.....	27
5.1 <i>Felhasználói felület</i>	27
5.1.1 Főoldal	27
5.1.2 Regisztráció	29
5.1.3 Bejelentkezés után	30
5.2 <i>Adminisztrátori felület</i>	33
5.2.1 Termékek felvétele	33
5.2.2 Termékek törlése/módosítása	34
5.2.3 Képfeltöltés.....	34
5.2.4 Felhasználók törlése	35
5.2.5 E-mail listakarbantartás	35
5.2.6 Rendelések.....	35
5.2.7 <i>E-mail küldése</i>	36
5.3 <i>Biztonsági és technikai megoldások</i>	37
5.3.1 Lekérdezések	37
5.3.2 A kereső	39
5.3.3 PHPmailer függvénykönyvtár	39
5.3.4 Adatok átadása.....	41
5.3.5 CAPTCHA	41
5.3.6 HTTP-hitelesítés.....	43
5.3.7 Az SHA() függvény.....	44
5.3.8 A sütik	44
5.3.9 A szabályos kifejezések vizsgálata.....	45
6. Összefoglalás	47
7. Irodalomjegyzék	49
8. Köszönetnyilvánítás.....	50

1. Bevezetés

A XXI. században az informatika nélkül szinte elképzelhetetlen az élet. Az emberek nagy része nap, mint nap használja az informatikai termékeket a munkahelyén, vagy szórakozás céljából. Az informatikával együtt az internet is szintén rohamosan fejlődő és egyre szélesebb körben használt technológia. Az elmúlt harminc évben nem várt fejlődést produkált minden az informatikával kapcsolatos iparág, így az online vásárlás is. Hazánk online iparágának történelme valamikor az 1990-es évek közepére nyúlik vissza, amikor elindult a telekommunikációs infrastruktúra erőteljes fejlesztése, az internet sávszélesség robbanása. Ez a folyamat az interneten elérhető magyar nyelvű tartalom mennyiségi gyarapodásában és az internet felhasználók számának növekedésében nyilvánult meg.

Egyes kutatások szerint 2010-ben a 15-69 éves lakosság 50-55 százaléka internetezik legalább havi egy alkalommal Magyarországon ez körülbelül három millió embert jelent. Közülük legalább 50 százalék napi szinten találkozik az internettel és ennek a másfél millió embernek legalább a fele már több alkalommal, mondhatni rendszeresen vásárolt már online. Ugyan a magyarországi emberek igen nagy százalékban bizalmatlanok az online vásárlással kapcsolatban, így is minden évben legalább húsz százalékos bevétel növekedést könyvelhettek el, az ebben a szférában jelenlevő cégek. Köszönhető ez az új generációnak, aminek tagjai már az informatikával körülvéve nőttek fel és így abba vetett bizalmuk is jóval nagyobb.

A szakdolgozatom témaválasztása azért esett épp a webes alkalmazásfejlesztésre mivel, véleményem szerint az elkövetkezendő húsz évben majdnem mindenkinek, aki életképes üzletet kíván üzemeltetni, meg kell jelennie az online piacon is. Illetve a háttérben használatos informatikai megoldások is régóta foglalkoztatnak. Az alkalmazást a családom tulajdonába lévő papír-írószer bolt köré építettem fel, viszont kisebb módosításokkal akármilyen más típusú termékek online értékesítésére felhasználható lehet.

Célom:

- Az adatbázis kezelés alapjainak bemutatása
- A Web alapú alkalmazásokban rejlő lehetőségeinek feltárása
- A WAMP: Windows+Apache+MySQL+PHP négyes használatának megismertetése

Alkalmazásom fejlesztéséhez a PHP nyelvet választottam, mégpedig a:

- Stabilitása
- Platform függetlensége
- Egyszerű kezelhetősége miatt

A PHP programozása nagy előnyei közé tartozik, hogy nem igényel semmilyen speciális fejlesztői környezetet, megírásához elegendő akár a jegyzetomb is, tesztelése, futtatása pedig akármilyen böngészővel lehetséges. Az adatbázis kezeléséhez a MySQL-t használtam, amely egyszerű, gyors és megbízható adatbázis-kezelést tesz lehetővé a PHP programunkban.

Az alkalmazásom bemutatását a következő részekre bontva végzem:

- Felhasználói felület
- Adminisztrátori felület
- Biztonsági és technikai megoldások

Napjainkban a felhasználók elég jelentős hányada csak használja az alkalmazásokat és nincs igazán tisztában vele, hogy a háttérben milyen folyamatok zajlanak le. Célom, hogyha valaki elolvassa a dolgozatot, megértse az alkalmazás felépítését és működésének lényegét akkor is, ha esetleg a programozásban nem jártas személyről van szó. A felhasználói felület illetve az adminisztrátori oldalak bemutatása ebben a szellemben íródott meg, míg a harmadik rész, a technikai megoldások már feltételez némi informatikai alaptudást az olvasótól. A felhasználói részben a felhasználók által elérhető funkciókat mutatom be. Az adminisztrátori részben az alkalmazás által nyújtott adminisztrátori lehetőségeket vizsgálom meg, amelyekhez a speciális adminisztrátori jogosultsággal rendelkező felhasználók férhetnek hozzá. A technikai részben pedig betekintést nyerhet az olvasó a felhasznált technológiai megoldásokba, természetesen a teljesség igénye nélkül.

2. Technológiák ismertetése

2.1. HTML

2.1.1. A HTML rövid története

Az 1990 évek elején, Tim Berners-Lee elkészítette az első honlapot és web szerveret, megalapította a World Wide Web Consortiumot., és megalkotta a HTML nyelv első változatát. A nyelv sokat fejlődött az első napok óta, mégis az akkor definiált HTML tag-ek több mint fele most is létezik és a mostanában legfrissebb 5-ös számú specifikációban is felfedezhetőek.

Az IETF (Internet Engineering Task Force) a HTML ajánlásának első vázlatát 1993-ban adta ki. 1995-ben elkészült a HTML 2.0, amely sok ötletet merített az eredeti HTML ajánlásból. A HTML 3.0 ajánlása még ugyanebben az évben érkezett, de a böngészőgyártók más irányban fejlesztettek, így ez az ajánlás nem volt sikeres. 1997-ben a W3C kiadta ajánlásként a HTML 4.0-át, amelyben még több böngésző-specifikus kiterjesztést próbáltak meg ésszerűsíteni és egyszerűbbé tenni. Ezt azzal érték el, hogy több elemet elavultként jelöltek meg, ez azt jelenti, hogy ezek az elemek még léteznek ebben a verzióban, de a következőben már teljesen törölve lesznek.

A HTML 4.01 1999-ben jelent meg, majd néhány elírást javítottak benne 2001-ben. Ez a HTML sokáig működő verziója melyet a 2010-es HTML 5 ajánlás hivatott leváltani.

2.1.2. A HTML nyelv alapjai

A HTML (**H**yper **T**ext **M**arkup **L**anguage azaz Dokumentumleíró nyelv) alapelemei a címkék vagy angolul a tag-ek, a címkéken belüli adatokról a böngésző tudja, hogy milyen módon kell megjelenítenie.

Egy egyszerű példa tag-es felépítésre:

```
1 <HTML>
2 <HEAD>
3 <TITLE>Ez a dokumentum címe</TITLE>
4 </HEAD>
5 <BODY bgcolor=#006633>
6 <H1>Első rész</H1>
7 <H2>Első fejezet</H2>
8 <P>Ez itt egy bekezdés.</P>
9 <P align="center">Ez egy újabb,
10 középre igazítva!</P>
11 <P align="right">Ez pedig jobboldalt.</P>
12 <H6>Második fejezet</H6>
13 <H2>Harmadik fejezet</H2>
14 <H1>Második rész</H1>
15 </BODY>
16 </HTML>
17
```

Az első tag-gel <HTML> tudatjuk böngészőnkkel, hogy ez egy HTML kód lesz. A HEAD szokás szerint ez a második utasítás, mert a tartalmát a megjelenítők a dokumentumok elejére helyezik. Mint kezdő tartalom alkalmas indexek, katalógusok elkészítésére.

Bizonyos kódok csak a HTML dokumentum fejlécébe helyezhetők el: pl.: főcím (TITLE), indexáló (ISINDEX), bázis (BASE), azonosító (NEXTID), állománykapcsoló (LINK), metacím (META). Mindegyik tag-nek van záró tag-e ami csak abban különbözik a nyitótól, hogy a < jel után még van egy / jel is.

Főcím. <TITLE> ... </TITLE> jelek közé helyezve lehet beírni a kódba. A böngészők ezt az ablak címsorában jelentetik meg. A dokumentum címe mindenképpen érthető, kifejező cím kell, hogy legyen, mert a felhasználók sokszor találkoznak vele. A TITLE utasítás csak a HEAD utasításon belül helyezhető el, csak szöveget tartalmazhat, beágyazott utasítás nem lehet benne.

Címsorok, Angolul Heading. Hat hierarchikusan megjelenítendő alcím használatára van mód, amelyeket a H betű, és egytől hatig számok jelölnek: **H1**, **H2**, **H3** stb. Az első a leghangsúlyosabb alcím, a **H6**-os a legkevésbé. Nem tanácsos kihagyni lépcsőket, mert összezavarhatja az olvasót.

A **Bekezdés azaz** Paragraph. A <P> és </P> jelek közé írandó. A kezdő és befejező utasítás között található szöveget a megjelenítők bekezdésként értelmezik. Az utasítás paramétereiként megadható több formázási mód pl.: a szöveg elhelyezésnek helye, mérete, betűtípusa, mint az a példában is látható.

A példa megjelenítése böngészőben:



2.2. PHP

2.2.1. A PHP története

A PHP-t Rasmus Lerdorf alkotta meg 1994-ben. Ez a teszt verzió volt csak nem lett kiadva soha. Az első kiadott verzió "Personal Home Page Tools" nevet viselte. Egy nagyon egyszerű feldolgozó programból állt, ami csak néhány speciális makrót értett meg, valamint tartalmazott számos eszközt, amiket akkoriban gyakran használtak a honlapokon (számláló, vendégkönyv és hasonló). A feldolgozó program újraírása után 1995 közepén a "PHP/FI 2. verzió" nevet kapta. Ez a verzió mSQL-t is támogatta. A program népszerűségét mutatja, hogy 1996 végére már közel 15.000 honlapon alkalmazták a világon mindenütt. 1997-ben a feldolgozó programot Zeev Suraski és Andi Gutmans teljesen újraírták, és ez alkotta a lelkét az új PHP 3-nak. A legfrissebb változat (a PHP 5) a Zend (<http://www.zend.com/>) Scripting

Engine használatával nagyobb teljesítményt nyújt, még több külső könyvtárat és kiterjesztést támogat, és modulként képes futni minden népszerű webkiszolgálón. Most a PHP 4-es és a PHP 5-ös verzióját több mint öt millió szerveren használják.

2.2.2. A PHP nyelv alapjai

A HTML tökéletes, ha egy statikus weboldalt kívánunk létrehozni, ám ha szeretnénk a weblapot dinamikusabbá, interaktívabbá tenni akkor erre a megfelelő eszközt a PHP. A PHP (Hypertext Preprocessor) egy szerver oldali HTML-be ágyazott szkriptnyelv.

A HTML kódba egyszerűen beépíthetők a PHP kódok blokkjai. Speciális HTML karakterekkel ki és be lehet lépni a PHP részekbe. Ezek a karakterek általában `<?php` és a `?>`. Ebből tudja a kiszolgáló, hogy itt PHP kód következik. Az különbözteti meg a PHP-t például a kliens oldali JavaScript nyelvtől, hogy a kód a kiszolgálón fut. A PHP kód tárolható elkülönítve egy .php kiterjesztésű fájlban, vagy beépítve a HTML kódba, ám ha a HTML fájlba beillesztünk egy PHP tag-et akkor a fájlunk kiterjesztése .php-ra változik. A lenti ábrán látható egy egyszerű példa a PHP beillesztésére a HTML kódba.

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>Php próba</title>
6 </head>
7 <body>
8
9 <?php
10
11 print "Helló Világ";
12
13 ?>
14
15 </body>
16 </html>
```

A PHP tag elhelyezése a HTML kódon belül tulajdonképpen lényegtelen, akárhova írhatjuk ahol nekünk a legmegfelelőbb. Amikor a forráskódban elér a web szerver a PHP nyitójeléhez,

a példában a kilencedik sor, átadja a PHP fordítónak a vezérlést mindaddig, míg el nem éri a záró tag-et. A záró tag a tizenharmadik sorban található. A web szerver a Php értelmezőt felhasználva elkészíti a végleges kimeneti HTML kódot. Ez gyakran lekérdezéseket, és egyéb olyan parancsokat tartalmaz, amelyből a számítógép előtt ülő felhasználó semmit sem vesz észre. Ez példában is igaz, a számítógép előtt ülő felhasználó nem tudhatja, hogy a Helló világ szöveget a PHP fordító állította elő vagy a HTML kóddal írtuk ki a böngészőjével. Ha a felhasználó megvizsgálja, a böngészőben az oldal forráskódját ott sem találja semmi nyomát a PHP kódnak, mivel az ő számítógépe csak a szerver által feldolgozott PHP kód eredményét kapja meg, ami HTML formájában tűnik fel.

2.3. MySQL

2.3.1. A MySQL fejlődése

A SQL története egészen 1979-ig vezethető vissza, amikor Michael Widenius kifejlesztett egy adatbázis-kezelőt UNIREG néven a svéd TcX vállalat számára. Ez azonban nem rendelkezett SQL interfésszel, ezért az 1990-es évek közepére korszerűtlenné vált és a cég elkezdett különböző alternatívákat keresni. Ilyen volt az mSQL (vagy Mini SQL) nevű, kliens/szerver felépítésű rendszer, mely jól töltötte ki azt a rést, amely az asztali számítógépeken népszerű Microsoft Access, illetve a nagy erejű, vállalati szintű relációs rendszerek, pl. Oracle vagy IBM DB2 között húzódozott. Az mSQL sem felelt meg azonban a TcX számára, ezért Widenius egy új, saját rendszer kifejlesztése mellett döntött. A fejlesztés 1996 májusában fejeződött be. Az elkészült termék lett az első verziója annak, amit ma MySQL néven ismerünk. A My elnevezés Michael Widenius lányának nevéből adódik akit My-nak neveztek.

1995-ben Widenius két társával együtt megalapította Uppsalában a MySQL AB részvénytársaságot (az AB a svéd Aktiebolag rövidítése, a magyar Rt.-nek felel meg). A MySQL AB-t tekintik manapság az ún. „második generációs” nyílt forráskódú cégek egyik első reprezentánsának. Ami a második generációs társaságokat megkülönbözteti az első generációs nyílt forráskódú üzleti modelltől, az a kettős licencezés. A MySQL esetében a kettős licencezés lényege, hogy az adatbázis szerver kétféle módon áll rendelkezésre: Community (közösségi) Server és Enterprise (vállalati) Server változatban. A közösségi változat forráskódját bárki letöltheti, ingyen használhatja, sőt szükség esetén akár

módosíthatja is a GPL (GNU General Public Licence) szabályai szerint. Ugyanakkor a rendszer üzemeltetéséhez és karbantartásához nem jár a szoftver tulajdonosának támogatása, e téren a felhasználó magára van utalva. Az újabb verziók megjelenése nincs előre ütemezve. Előfordulhat, hogy két teljes (bináris és forráskód alapú) kibocsátás között, egyes verziók csak forráskód alakban állnak rendelkezésre. Azok a vállalatok illetve magánszemélyek, akik magasabb szintű támogatást igényelnek, előfizethetnek a vállalati szerverre. Az előfizetési díj ellenében (mely azonban e változat esetében is jóval alacsonyabb, mint az ismert kereskedelmi célú alternatívák licenc és karbantartási díja) rendszeres frissítés, hibajavítás és konzultációs szolgáltatás jár a MySQL mérnökei részéről. Az előfizető havonta kap frissítéseket (update) és negyedévenként ki-egészítő csomagokat (service pack). A két változat között csak a kibocsátás stratégiája és a szolgáltatások mennyisége eltérő. Mindkét változat, azonos verziószám esetében, ugyanazt a szerveret tartalmazza. A MySQL jelenleg tíz milliót jóval meghaladó telepítéssel rendelkezik. Az alkalmazások spektruma az egyéni kis felhasználóktól a legnagyobb cégekig terjed. Néhány ismertebb felhasználó: NASA, Yahoo, Linux.com, Slashdot.org, Nortel, Ericsson, Alcatel, Nokia, Motorola.

2.3.2. A MySQL jellemzői

A MySQL egy több felhasználós, többszálú, SQL-alapú relációs adatbázis-kezelő szerver. Egyedi illesztő felületekkel az adatbázis-kezelő elérhető C, C++, C#, Delphi, Eiffel, Smalltalk, Java, Lisp, Perl, PHP, Python, Ruby és Tcl programozási nyelvvel. Egy MyODBC nevű ODBC interfész további, ODBC-t kezelő nyelvek számára is hozzáférhetővé teszi az adatbázis-kezelőt. A MySQL számára az ANSI C a natív nyelv. A MySQL különböző platformokon futtatható: AIX, BSDi, FreeBSD, HP-UX, Linux, Mac OS X, NetBSD, 0Netware, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP és a Windows frissebb verziói. Már MacOSX 10.4-re is elérhető

A MySQL rengeteg hasznos funkcióval rendelkezik melyek közt sok olyan is van, amelyek csak a MySQL-ben található meg, számos más relációs adatbázisrendszerrel ellentétben.

- Többféle tároló motor, amelyek között bármely táblához szabadon választhatunk.
- Natív tároló motorok (MyISAM, Falcon, Merge, Memory, MySQL federated, MySQL Archive, CSV, Blackhole, MySQL Cluster, Berkeley DB, EXAMPLE, és Maria).
- Partnerek által fejlesztett tároló motorok (InnoDB, SolidDB, NitroEDB, BrightHouse)
- Közösségi fejlesztésű tároló motorok (memcached, httpd, PBXT, Revision Engine)
- Akár egyéni tároló motor
- Commitek csoportosítása, több tranzakció fogadása többféle kapcsolatról, melyek meggyorsítják a tranzakciók lefolyását

2.4. Mi az a CSS?

A már korábban említett formázási lehetőségek, amelyeket a HTML nyelv nyújtott egy bizonyos idő után már nem feleltek meg a fejlesztők és az internet nagyközönsége számára. Ezért Hakon Wium Lie 1994-ben úgy döntött létrehoz egy nyelvet amely nagyobb szabadságot biztosít a felhasználók számára. Ehez Bert Bos-tól kapott segítséget aki időközben egy Argó nevű böngészőn dolgozott, mely saját stíluslapokat használt, így végül ők ketten döntöttek a CSS kifejlesztése mellett. A CSS jelentése Cascading Style Sheets, azaz egymásba ágyazott stíluslapok, a számítástechnikában egy stílusleíró nyelv, mely a HTML vagy XHTML típusú strukturált dokumentumok megjelenését írja le. Ezen kívül használható bármilyen XML alapú dokumentum stílusának leírására is, mint például az SVG, XUL stb.. A CSS-t a weblapok szerkesztői és olvasói egyaránt használhatják, hogy átállítsák vele a lapok színét, betűtípusait, elrendezését, és más megjelenéshez kapcsolódó elemeit. A tervezése során a legfontosabb szempont az volt, hogy elkülönítsék a dokumentumok struktúráját (melyet HTML vagy egy hasonló leíró nyelvben lehet megadni) a dokumentum megjelenésétől (melyet CSS-sel lehet megadni). Az ilyen elkülönítésnek több haszna is van, egyrészt növeli a weblapok használhatóságát, rugalmasságát és a megjelenés kezelhetőségét, másrészt csökkenti a dokumentum tartalmi struktúrájának komplexitását.

Az elemek stílusát különböző CSS szelektorokkal lehet kiválasztani:

- Minden elemre – a * szelektor használatával
- Az elem neve alapján – például minden 'p' vagy 'h2' HTML elemre
- Leszármazottak alapján – például az olyan 'a' elemekre, melyek egy 'li' elem részei, a szelektor "li a"
- class vagy id attribútumok alapján – például .class és/vagy #id a class="osztály" vagy id="azonosító" elemekhez

Ezekon kívül rendelkezésre áll több pszeudo-osztály, melyekkel további műveletekhez lehet stílust rendelni. Talán a legismertebb ezek közül a :hover, melynek stílusa akkor lép érvénybe, mikor a hozzá tartozó elem aktiválódik, például fölé visszük az egeret. Hozzá lehet fűzni a szelektorokhoz is, például a:hover vagy #element id:hover. További ismertebb pszeudo-osztályok a :first-line, a :visited vagy a :before.

A CSS használatának legfontosabb előnyei:

- Több lap vagy akár egy teljes webhely stílusait egy helyen lehet tárolni, így gyorsan és könnyen frissíthető
- Különböző felhasználókhöz különböző stílusokat lehet rendelni.
- A dokumentum mérete csökken, mivel nem tartalmaz információkat a megjelenítéshez

A CSS egyszerű szintaxissal rendelkezik, csak néhány angol nyelvű kulcsszót használ a stílusok tulajdonságaihoz. A stíluslap maga a stílust leíró szabályok sora. Minden szabályhoz tartozik egy szelektor és egy deklarációs szakasz. Ez utóbbi kapcsos zárójelek között pontosvesszővel elválasztott deklarációkat tartalmaz. A deklarációk formája a következő: a tulajdonság neve, egy kettőspont, majd az adott tulajdonság értéke.

Például:

```
4 p {
5   font-family: "Garamond", serif;
6 }
7 h2 {
8   font-size: 110%;
9   color: red;
10  background-color: white;
11 }
12 .megjegyzes {
13   color: red;
14   background-color: yellow;
15   font-weight: bold;
16 }
17 p#bevezetes {
18   border: 3px dashed black;
19 }
20 a:hover {
21   color: yellow;
22   background-color: navy;
23 }
```

Ez összesen öt szabály, a p, h2, .megjegyzes, p#bevezetes és a:hover szelektorokra. A deklarációra egy példa a color: red, ahol a color (szín) tulajdonság értéke red (vörös). Az első két szabályban a p (bekezdés) és h2 (második szintű címsor) HTML elemekhez rendeltünk stilisztikai tulajdonságokat. A bekezdések Garamond betűtípussal lesznek megjelenítve, vagy ha nincs ilyen, akkor valamilyen más talpas (serif) betűtípussal. A második szintű címsorok vörös színnel és fehér háttérrel jelennek meg. A harmadik szabály egy CSS osztályt definiál, melyet bármilyen elemhez hozzá lehet rendelni a dokumentumban a class attribútummal, például:

<p class="megjegyzés"> Ez a bekezdés félkövér, vörös színű lesz, sárga háttérrel.</p>

A negyedik szabály egy olyan p elemre vonatkozik, melynek id attribútuma bevezetes: ez az elem egy 3 pixel széles fekete szaggatott keretet fog kapni. Az utolsó szabály a hover állapothoz tartozó megjelenést szabályozza az elemhez (többek között a hiperlinkek is ilyen elemek). A szabály meghatározza, hogyan nézzen ki az elem, amikor az egérmutató fölötte áll. A példában az elem háttérszíne ilyenkor sötétkék, a szövege színe sárga lesz.

2.5. JavaScript

A JavaScript programozási nyelv, amelyet weblapokon elterjedten használnak. Egy objektum orientált szkriptnyelv. Szülőatyja Brendan Eich, a Netscape Communications mérnöke neve először Mocha, majd LiveScript volt, később „JavaScript” nevet kapott, és szintaxisa közelebb került a Sun Microsystems Java programozási nyelvéhez. A JavaScriptet először 1997–99 között szabványosította az ECMA „ECMA script”néven. A jelenleg is érvényes szabvány az ECMA-262 Edition 3 (1999. decemberi), ami a JavaScript 1.5-nek felel meg. Ez a szabvány egyben ISO szabvány is. A Java programozási nyelvhez nem csak a nevében, hanem szintaxisában is megegyezik. A legfontosabb különbség az, hogy a Java forráskódhoz szükség van fordítóra, hogy bájtkódot készíthessünk, míg a JavaScript egy parancsnyelv, aminek futtatásához elegendő egy böngésző. A JavaScript kódot a HTML fájlban, vagy külön egy általában *.js kiterjesztésű fájlban szokás tárolni. A JavaScripttel dinamikussá tehetjük honlapunkat. Egyrészt látványosabbá tehetjük vele azokat (dinamikusan, futásidőben generálhatunk HTML lapokat), másrészt számos praktikus dologra használhatjuk fel (pl. jelszót kérhetünk a lapunk megtekintéséhez, vagy pl. ellenőrizhetjük az elküldendő adatok helyességét még a kliens oldalon - hálózati erőforrásokat spórolva meg, stb).

A JavaScript utasításoknak a SCRIPT elemben kell lenniük az alábbi módon:

```
<SCRIPT>
```

```
JavaScript-utasítások
```

```
</SCRIPT>
```

A scriptben lévő utasításokat csak azon böngészők fogják végrehajtani amelyek, ismerik a JavaScriptet. A forrásprogram szövegét célszerű egy HTML kommentbe elhelyezni a JavaScriptet esetleg nem ismerő böngészők miatt. A használatos aritmetikai és relációs operátorok valamint a logikai operátorok megegyeznek a Java nyelvben használatosakkal. A változók deklarációjának kulcsszava a var. A változó neve:

- Csak betűvel kezdődhet és
- Nem egyezhet meg a JavaScript foglalt szavaival.(if, then, for, next)

A változó neve akármilyen hosszúságú lehet. A JavaScript érzékeny a kis és nagybetűk közti különbségre. A változónak hatféle adattípusa lehet:

- Számtípusú, ha értékadásnál számadattípus értéket vesz fel.
- String típusú, ha szöveges adatokkal töltjük fel.
- Boolean típusú, ha true vagy false értéket vesz fel.
- Null adattípusú.
- Undefid változó, amely még nem kapott értéket.
- Objektum.

A nyelv vezérlőszerkezetei a következők

- Feltételes utasítások pl.: if, else.
- Elágazások pl.: swich , case.
- Ciklusok pl.: for , do... while.

A nyelv ezen a téren is megegyező a Java nyelvvel. A függvények meghívása pedig a következő képen alakul:

```
function function_neve(argumentumok){  
utasítások  
}
```

3. Felhasznált eszközök

3.1. Marcomedia Dreamwiewer 2004 MX

Az Adobe cég rengeteg kiváló terméke közül választottam honlapom fejlesztéséhez a Marcomedia Dreamwiewer egy régebbi változatát. A legfőbb ok az, hogy rendelkezem egy jogtisztá példánnyal ebből a szoftverből. A kiadás éve 2003 , hiába számít már idősebb szoftvernek a mai termékekkel is fel tudja venni a versenyt. A programnak kiváló a felépítése egyszerű kezelni egy avatatlan kéz számára is. A szerkesztő felület jól átlátható, színekkel különbözteti meg a különböző kódrészeket, valamint a legkényelmesebb funkciója a vizuális szerkesztő felület. Itt egy kattintással megnézhetjük a létrehozott változások eredményeit. A program ismer minden fontosabb nyelvet (HTML, XML , JavaScript, PHP , CSS), így nem kell többféle szerkesztőt használnunk egy fejlesztés során.

3.2. WampServer

A WampServer egyszerre tartalmazza a három valószínűleg legnépszerűbb és leggyakrabban használt szoftvert, amelyek a dinamikus oldalak létrehozásánál segítik a programozót. Ez egy tesztkörnyezet, amelyben a saját számítógépünkön kipróbálhatjuk a készített kódot, nem kell mindig feltöltenünk kiszolgálókra és úgy ellenőrizni. Óriási mértékben gyorsítja a fejlesztési folyamatot. Ezt a hármast szokták LAMP mozaikszóval is jelölni. Ebben az esetben viszont van egy negyedik tag a sor elején a Linux. Én Windows alatt készítettem a honlapot és az Apache, MySQL valamint a PHP ugyanúgy hibátlanul működött nálam is. A telepítéssel és a beállításokkal így sokat nem kell foglalatzkodni, hisz a program egymás után fellelepít és beállít mindent. Elindítása után egy az értesítési területen megjelenő ikonra kattintva egy egyszerű menüből kezelhetjük az összetevőket egy helyen. Itt hozzáférhetünk a .ini és .conf fájlokhoz. Valamint a <http://localhost> és a phpMyAdmin is elérhető egyetlen mozdulattal.

Jelenleg a 2.0-ás változatát használom, amely tartalmazza az:

- Apache 2.2.11
- PHP 5.3.0
- MySQL 5.1.36

3.3. PHPMYAdmin

Egy könnyedén kezelhető felületet nyújt adatbázisunk létrehozásához, karbantartásához, módosításához. Nem kell mindig külön parancsokkal kommunikálni a PHP szerverrel, hanem egyszerűen egy böngésző ablakában egérrel végrehajthatunk lekérdezéseket, változtatásokat, amelyek parancssorból igen körülményesen megoldhatók. A szoftver ingyenesen hozzáférhető és rendkívül gyorsítja a táblákkal való munkánkat, valamint a grafikus felület segítségével könnyebben áttekinthetőek még a bonyolult táblaszerkezetek és kapcsolatok is. Az adatbázisra kattintva felső menüsorban lekérdezhető az adatbázis szerkezete, az SQL menüpontban lekérdezéseket futtathatunk a táblákon, valamint kereshetünk az adatbázisban, exportálhatjuk, importálhatjuk az adatbázisokat.

4. Az Adatbázis

Akármilyen alkalmazást is szeretnénk létrehozni rendkívül fontos az, hogy a háttérben működő adatbázisunk megfelelő módon legyen felépítve. Mielőtt neki állunk az alkalmazás kivitelezésének át kell gondolnunk, hogy miképpen álljon össze. A későbbi módosítások akár nagyon sok munkával is járhatnak. Adatbázisom igen kisméretű mégis gondot okozott, mikor módosítottam, bővítettem, ekkor szinte minden lekérdezésemet újra át kellett írni. Egy nagyméretű adatbázisnál, amihez tartozik egy több ezer kódsorból álló alkalmazás is, ez napokkal visszavetheti a munkát.

4.1. Adatbázis alapfogalmak

A **táblák**: A táblák az adatbázisunk legnagyobb alkotórészei. Összefüggő adatstruktúrák tárolására használjuk. A MySQL rendszerekben egy tábla létrehozásakor három fájl jön létre. Ezek a tábla meghatározását, a tárolt adatokat és az adatokhoz tartozó indexeket tároljuk, ugyanakkor egységként kell kezelni őket.

A **mezők**: A mezők adják a tábla szerkezetét, és határozzák meg a benne tárolt adat típusát, annak hosszát, ezért a mezők létrehozása a tábla létrehozásával egy időben történik. Lényegében egy táblázatkezelő oszlopainak lehetne megfeleltetni őket. Egy táblában pontosan 3398 mező lehet maximálisan.

A **recordok**: A record egy bejegyzés a táblában. Azt az értéket tartalmazza, amelyet valójában tárolni szeretnénk. A recordok lehetnek teljesen kitöltött, és kitöltetlen recordok, ami nem biztos, hogy üres recordot jelent. Mivel a mezőkhöz típustól függően alapértelmezett értéket állíthatunk be. Ha nem töltjük ki az adott mezőt, ez az alapértelmezett érték fog szerepelni a táblázatunkban.

A **kulcsok**: A kulcsok a táblák közötti kapcsolatok megteremtésének eszközei. A MySQL kétféle kulcstípust ismer, az **elsődleges** (un. primary key) és az **egyedi** (unique key) **kulcsokat**. Az elsődleges kulcsnak minden táblában szerepelnie kell. Ez az az érték, amellyel a kapcsolatokat meghatározzuk. Alapértelmezés szerint ez az érték is egyedi, valamint a gyors

működés miatt numerikus adatnak kell lennie. Nem tartalmazhat NULL értéket se. Az egyedi kulcsok megadása nem kötelező, de nagyon hasznos lehet, míg elsődleges kulcsból csak egy lehet táblákként, addig akárhány unique key létezhet.

4.2. A táblák létrehozása

Az adatbázis az alkalmazás jellege miatt kevés táblából áll:

Az adatbázisomban elhelyezkedő táblák:

- bolt_termekek
- bolt_user
- bolt_vasarlasok
- email_list

A PHPmyAdmin-ba belépés után egy könnyen átlátható, egyszerűen használható felületet láthatunk. A adatbázis létrehozása itt a kezdőoldalon megtehető csak be kell írunk a nevét és kiválasztani az illesztés fajtáját, majd megnyomni a létrehoz gombot. A következő megjelenő oldalon láthatjuk, hogy jelenleg üres az adatbázis és képernyő alsó részén megjelenik az új tábla létrehozása menüpont ahol meg kell adnunk a tábla nevén kívül, hogy hány mezője legyen az adott táblának. Ha innen tovább lépünk megjelenik egy űrlap, melyet feltölthetünk a mezőkre jellemző információkkal. A mező nevét, típusát, hosszát, alapértelmezett értékét, és egyéb tulajdonságait. Valamint az utolsó oszlopban még megjegyzést is írhatunk a mezőkhöz. Ha esetleg már előre meg van írva az SQL forrásfájl, akkor gyorsabban fel lehet vinni a táblákat az adatbázisba. Ha rákattintunk az általunk kívánt adatbázis nevére a baloldalon felsoroltak közül. Megjelenik egy felső menüsor ahol az importálásra kattintva, a már meglévő fájlból feltölthetjük az adatbázis tábláit, azok mezőit. Tábláim létrehozásához én ezt az utóbbi módszert választottam.

4.3. A táblák szerkezete

1, bolt_termekek

Ebben a táblában a termékek nevét és a hozzájuk tartozó egyéb információkat tároltam. A tábla mezői:

a, t_id : Ez a mező a termékazonosítója tehát egyedinek kell lennie és sosem vehet fel NULL értéket. Típusa INTEGER és a mezőn auto increment tulajdonság van beállítva azaz automatikusan sorban töltődik fel az értéke.

b, t_kategoria: Ebben a mezőben tárolódik egy INTEGER típusú érték amely attól függően változik, hogy a termék mely csoportba tartozik. Az értékét 1-4 –ig veheti fel, ha 1 papír-írószer ,ha 2 ajándék ha 3 nyomtatvány ha pedig 4 az érték akkor irodai eszköz a termék fajtája.

c, t_nev: A termék nevét tároló mező VARCHAR(50) típusú mező, értéke nem lehet NULL.

d, t_ar : A mező a termék árát tartalmazza.

e, t_leiras : Ebben a mezőben a termékekről részletes leírás adható melyet a honlapon a termék nevére kattintva megtekinthet a vásárló. A text típusú mező lehetővé teszi a hosszabb leírások tárolását is. Legfeljebb 65535 karakter tárolható a mezőben.

f, t_marka : A termék gyártójáról, márkájáról tartalmaz információkat

g, kep: Minden megvásárolható termékhez feltölthető kép is amely segít a vásárlónak jobban megismerni a terméket. A képet maga nincs adatbázisban, csak arra egy hivatkozás van tárolva, a kép a webtárhelyre van feltöltve.

2, bolt_user

A tábla azoknak a felhasználóknak az adatait tárolja akik regisztráltak az oldalra vásárlás céljából. A tábla mezői :

a, user_id: A mezőben egy INTEGER típusú érték tárolódik amely a regisztrált tagot egyedileg azonosítja, tehát értéke nem lehet NULL és ez a mező is auto increment tulajdonságú.

b, user_name: A felhasználó bejelentkezéshez használt neve, nem egyezhet másik felhasználó nevével. VARCHAR(50) típusú mező.

c, user_pass : A felhasználó által megadott jelszót tároljuk ebben a mezőben SHA titkosítással. VARCHAR(40) típusú mivel a titkosított jelszó mindig negyven karakter hosszúságú.

d, user_email: A felhasználó által megadott e-mail címet tartalmazza, aminek szintén egyedinek kell lennie, nehogy egy e-mail címről többen regisztráljanak.

e, vez_nev: A felhasználó vezetéknévét tárolódik benne.

f, ker:nev: A felhasználó keresztnévét tárolására szolgáló mező.

g, ir_szam: INTEGER típusú mező. A felhasználó irányítószámát tartalmazza.

h, varos: A felhasználó lakóvárosának nevét tartalmazza. Nem lehet NULL.

i, cím: A felhasználó pontos címét tárolja: utca, házszám, emelet, ajtó..

j, tel_szam: A felhasználó telefonszámát tartalmazó INTEGER típusú mező.

3, bolt_vasarlasok

A tábla a felhasználók vásárlásainak részleteit tárolja. A tábla mezői:

a, rendelesiszam : Egy véletlen generált szám, amely így egyedileg azonosít minden vásárlást, akkor is ha egy vásárláson belül több terméket rendel egyszerre a felhasználó.

b, vasarlasok_id: Ez a mező azonosítja a vásárlásokat, hogy egyszerűen egy darab sorszámmal lehessen hivatkozni külön-külön minden vásárlásra. Típusa INTEGER.

c, user_id: A bolt_user táblából átvett értéket tartalmazza, a vásárló pontos azonosítására szolgál. Értéke így nem lehet NULL, mivel az már a bolt_user táblában is ki lett kötve.

d, t_id: A bolt_termekek táblából átvett értéket tartalmazó mező, amely a vásárolt termékre hivatkozó sorszámot tárolja. Értéke így szintén nem lehet NULL.

e, statusz: A statusz mező egy darab számot tárol, ami mutatja a vásárlás jelenlegi státuszát. Egyestől hármásig vehet fel értéket. Az egyes érték automatikusan tárolódik mikor a felhasználó kosárba helyezi a terméket. A kettes érték azt jelenti, hogy a terméket megvásárolta a felhasználó. A hármas pedig azt jelenti, hogy a termék átkerült a régebbi vásárlások kategóriába.

f, db: A felhasználó által megvásárolt termék mennyiségét jelenti darabszámban. INTEGER típusú a mező.

g, teljesített : A mező nullát vagy egyet vehet fel értékül, ha nullás akkor azt jelenti, hogy a termék házhozszállítása még nem teljesült ha egyes akkor pedig igen.

h, datetime : timestamp típusú mező vásárláskor automatikusan feltöltődik, a vásárlás dátumát és pontos idejét hivatott tárolni.

4, email_list

A tábla a levelezési listára feliratkozott emberek adatait tárolja. A tábla mezői:

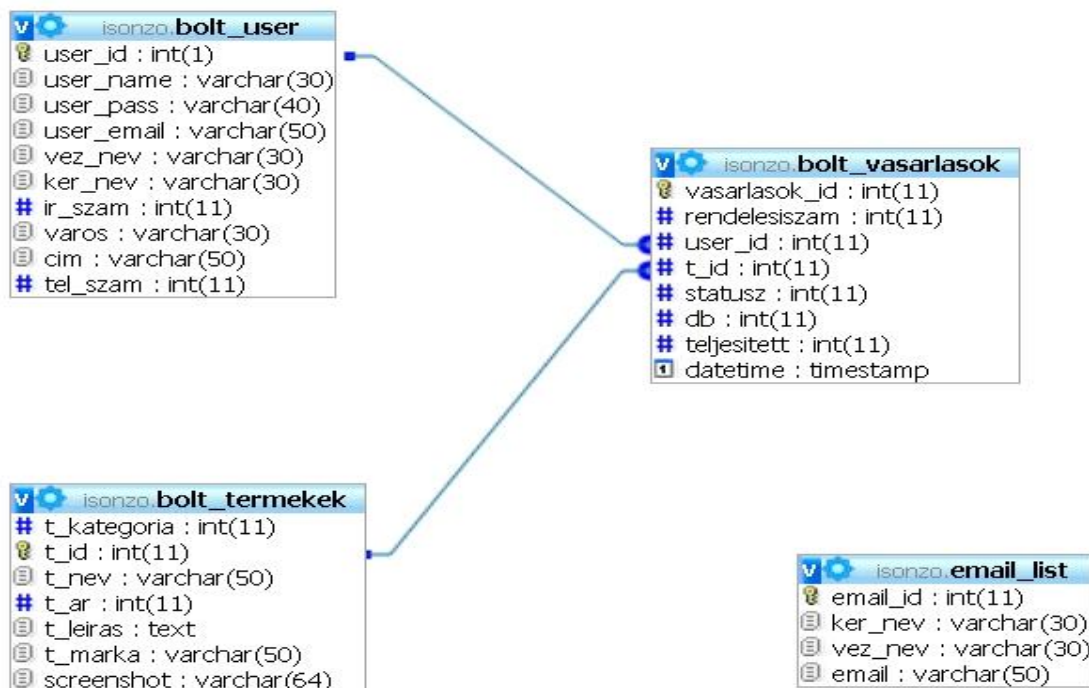
a, email_id: Egyedi azonosító számot tartalmazó mező, segítségével könnyedén hivatkozhatunk a tábla egy sorára. Értéke nem vehet fel NULL értéket és automatikusan inkrementálódó értéket vesz fel.

b, ker_nev: A hírlevélre feliratkozott felhasználó keresztnévét tároló mező, nem lehet üres mező.

c, vez_nev: A felhasználó vezetéknévét tartalmazza. Értéke nem lehet NULL.

d, email: A hírlevélre feliratkozott személy e-mail címét tartalmazó mező, értéke ennek sem vehet fel NULL-t.

4.4. A táblák kapcsolata



A táblák három féle kapcsolatban állhatnak egymással. Az első **egy az egyhez kapcsolat (1:1)**. Az egyik adattábla egyik rekordja a másik tábla egy rekordjához kapcsolódik. Vagy lehetséges **egy a többhöz kapcsolat (1:n)**. Ebben a kapcsolat típusban az egyik tábla egyetlen rekordja a másik tábla több rekordjához kapcsolódik. Végül a **több a többhöz kapcsolat (n:m)** az egyik adattábla egyik rekordja a másik adattábla több rekordjával van kapcsolatban és a másik adattábla egy rekordja az első adattábla több rekordjával áll kapcsolatban. Esetemben a mindkét kapcsolat egy a többhöz kapcsolat mivel egy felhasználó a bolt_user táblában csak egyszer szerepelhet és vásárolhat több terméket is így több sorban jelenik meg a bolt_vasarlasok táblában. Ugyanez igaz a termékazonosítóra is.

5. Az alkalmazás felépítés

5.1. Felhasználói felület

5.1.1. Főoldal

Ha a böngésző parancssorába begépelem a <http://localhost/> címet a következő oldal fogad, ez az alkalmazásom kezdőlapja az index.php. Az oldal automatikusan betöltődik mivel a böngészők egy cím beírása után keresik a szerveren lévő index nevű kezdőoldalt. Ez a cím csakis azon a számítógépen működik, ahol a fejlesztést végezzük, mivel a Wampserver alkalmazás csak egy tesztkörnyezet, ami csak a fejlesztő gépén fut. Az oldal a világhálón nem elérhető mivel, nincs feltöltve működő szerverre.

The screenshot shows the homepage of 'Isonzó Papírbolt'. At the top, there is a green header with the company name 'Isonzó Papírbolt' and the tagline 'Minden ami írószer'. Below the header, the date 'Március 30., 2011' is displayed. The main content area is divided into three columns. On the left, there is a search bar with the text 'Keresés a termékek között' and a 'Keresés' button. Below the search bar is a vertical menu with five items: 'főoldal', 'elérhetőségek', 'termékek', 'regisztráció', and 'rendelés'. The middle column features a section titled 'Rólunk' with two paragraphs of text. The right column contains logos for '3M', 'NYOMELL', and 'pátria nyomda', along with a 'pátria partner' logo. Below these logos is a 'Hírlevélre Feliratkozás' form with fields for 'Vezetéknév', 'Keresztnév', 'Email', and 'Ellenőrzőkód', and a 'Mehet' button.

Az oldal bal felső részén az aktuális dátumot megjelenítő rész található. Alatta helyezkedik el egy kereső mellyel a bolt által forgalmazott termékek között kereshetünk. A kereső alatt található a menüsor öt darab menüpontra. A középen elhelyezkedő rövid bemutatkozástól jobbra egy pár fontosabb beszállító és forgalmazott termék logója található. Ezek alatt pedig egy hírlevélre feliratkozás nevű panel, ahol nevünk és e-mail címünk megadásával feliratkozhatunk az oldal üzemeltetői által küldött e-mailek fogadására. Az oldal alján bal

oldalt egy link látható melyre kattintva felugrik az alapértelmezett levelezőprogramunk. A címzett és a tárgy mező alából kitöltve található meg, így a levél írójának csak a szövegmezőt kell kitöltenie a cégnek szánt üzenetével. A főoldal menüpont alatt található az elérhetőségek nevű menüpont, amelyben a BT. minden elérhetősége megtalálható a cím, telefonszám, a BT székhelye és faxszáma. Az oldal alján elhelyezett kép linkje pedig a googlemaps.com oldalra mutat ahol megtekinthető a bolt pontos elhelyezkedése és hozzá vezető útvonalak. Módunkban áll megtekinteni az összes, a bolt által forgalmazott terméket is a termékek menüpont alatt. Itt táblázatos formában felsorolva találhatóak az árucikkek. Több táblázat is van, mivel a termékek kategóriákra bontva jelennek meg ezen belül pedig ábécé sorrend van beállítva a könnyebb átláthatóság érdekében. A következő két menüpont a regisztráció és a rendelés. A regisztrációs oldalon van lehetősége a felhasználónak egy érvényes felhasználónév jelszó párost létrehozni. Amennyiben ezt nem teszi meg a rendelés menüpontba nem tud belépni mivel ahhoz, hogy valaki vásárolhasson be kell jelentkeznie az oldalra. Az összes a felhasználók által elérhető oldalon a menüsor illetve az oldal felső része ugyanúgy néz ki ezért a kód méretének csökkentése érdekében ezek a kódok a `fomenu.php` nevű fájlban vannak tárolva és minden oldalon ahol felhasználom őket, egy egysoros kóddal vannak helyettesítve, ami meghívja a `fomenu.php`-t. Ez a PHP kód a `require_once()`. Segítségével nem csak a kódok mérete csökken, de ha módosítani szeretnénk az oldal kinézetét illetve menüsorát, akkor csak egyetlen helyen kell átírni a kódot. Ez egy hasznos és kényelmes megoldás.

5.1.2. Regisztráció

Tehát mint már fentebb említettem a vásárláshoz szükség van felhasználónév és jelszó létrehozásához. Ha ezzel nem rendelkezünk a regisztráció menüpontra kattintva létrehozhatunk magunknak egyet.



The screenshot shows a registration form on a website. At the top, there is a banner with colorful paper clips and the text "Minden ami írószer". Below the banner, the date "szeptember 30, 2011" is visible. The main content area is titled "Felhasználó létrehozása" and contains the following text: "A vásárláshoz kérem regisztráljon, vagy ha rendelkezik érvényes felhasználónévvel jelentkezzen be." The form itself is enclosed in a box and has the following fields: "Regisztráció" (header), "Felhasználónév:" (text input), "Jelszó:" (text input), "Jelszó újra:" (text input), "e-mail cím:" (text input), and "ellenőrzőkód:" (text input with a "Írja be a kódot." label). To the right of the "ellenőrzőkód:" field, there is a CAPTCHA image showing the text "gcg11r". Below the form is a "Mehet" button. On the left side of the page, there is a navigation menu with items like "Főoldal", "Elérhetőségek", "Termékek", "Regisztráció", and "Rendelés".

A regisztráció során meg kell adni egy felhasználónevet, jelszót, e-mail címet, valamint egy ellenőrző kódot, amely segít megakadályozni a regisztrációs robotok kártételét. Ha felhasználónév nem megfelelő formátumú vagy már esetleg létezik a rendszerben ilyen névvel felhasználó, akkor hiba üzenetet kapunk. A névnek legalább hat karakterből kell állnia és nem kezdődhet csak betű karakterrel, a további karakterek lehetnek számok illetve betűk is, a felhasználó igényeinek megfelelően. Szintén hibát eredményez, ha nem töltünk ki minden mezőt maradéktalanul, vagy a jelzett formátumnak megfelelően. A jelszó is minimum hat karaktert kell, hogy tartalmazzon, ezek a karakterek lehetnek számok és betűk vegyesen tetszőleges formában. Az oldal kis- és nagybetű érzékeny. Abban az esetben, ha minden rendben ment a regisztráció során a következő üzenetet láthatjuk megjelenni a képernyőn: Sikeres regisztráció. Bejelentkezés után [itt](#) válogathat termékeinkből. A regisztrációról e-

mail-t is kap a felhasználó, mely tartalmazza a felhasználónevét és jelszavát emlékeztetőül. Az [itt](#) linkre kattintva a felhasználó első alkalommal bejelentkezhet és megkezdheti a vásárlást vagy módosíthatja adatait, esetleg megadhatja a többi a vásárláshoz, kiszállításhoz elengedhetetlen adatokat, az adataim menüpontban.

5.1.3. Bejelentkezés után

A felhasználó belépés után a következő oldalt látja maga előtt.



Isonzó Papírbolt
Minden ami írószerszám

December 29, 2010

bejelentkezés mint zowert

Kijelentkezés

Adataim

Kosaram

Régebbi vásárlásaim

A zowert nevű felhasználó törlése

Törlés

Termék Vásárlása

Irodai Termékek

A kosárba helyezéshez válassza ki a terméket és nyomja meg a kosárba gombot

Név	Csoport	Leírás	Márka/Gyártó	Ár	
PVC gyorsítűző	Irodai	-	-	70 Ft	Részletek
Írártartó	Irodai	gyűrűs írártartó	Silver Ball	550 Ft	Részletek
Átíró tömb	Irodai	A/5 50 x 2	-	380 Ft	Részletek
Tűzőgép	Irodai	Sax 620	Sax	1990 Ft	Részletek
Tűzőgép	Irodai	lto 100	Navus	960 Ft	Részletek
1-240	Irodai	leporellő	TICO	3800 Ft	Részletek
2-240	Irodai	leporellő 900 L	TICO	5200 Ft	Részletek
Cellux	Irodai	10 mm es	-	40 Ft	Részletek
Cellux	Irodai	19 mm	-	75 Ft	Részletek
Csomagolópapír	Ajandék	50 x 70	OFFSZET	95 Ft	Részletek
Csomagolópapír	Ajandék	50 x 70 Krepp	-	70 Ft	Részletek
Csomagolópapír fólia	Ajandék	70 x 90	-	190 Ft	Részletek

A kezdő oldalon rögtön vásárolhatunk is, minden termék, ami elérhető a boltban egy táblázatban jelenik meg, melyben fel van sorolva a termék neve, csoportja, gyártója, ára és pontos leírása. Minden sor végén van egy részletek link, amire rákattintva megjelenik egy külön oldalon egy részletes leírás, egy a terméket ábrázoló képpel együtt. A felhasználó ezen

az oldalon pontos tájékoztatást olvashat a termék részleteiről, gyártójáról, és az ára is fel van tüntetve. Az alsó részen pedig egy kosárba feliratú gomb, amivel a terméket behelyezhetjük virtuális bevásárló kosarunkba. A gomb előtt egy rubrika ahova meg kell adnunk az adott árucikkből hány darabot szeretnénk megvásárolni. Ez az érték alapértelmezetten egy, de a vásárló módosíthatja igényeinek megfelelően. A bal felső sarokban a dátum mellett található egy vissza feliratú gomb, amennyiben a vásárló tetszését mégsem nyerte el a termék, ezzel a gombbal léphet vissza az előző oldalra, ahol folytathatja a nézelődést.

A bal oldali menü legfelső részén van feltüntetve, milyen névvel vagyunk bejelentkezve. Szerepel még az adataim menüpont ahol a felhasználó személyes adatait módosíthatja vagy egészítheti ki. Az oldalon egy űrlap jelenik meg, ahova a már felvitt adatok be vannak írva. A többi mezőt is ki kell töltenie a felhasználónak, ha ténylegesen vásárolni szeretne. Ezen kívül megtekinthetjük régebbi már megvásárolt termékeinket is, ha már vásároltunk az üzletben. A kijelentkezés menüponntal elhagyhatjuk felhasználói fiókunkat.

Esetleg ha szeretnénk törölni magunkat a rendszerből az a törlésre kattintva tehetjük meg. Ekkor az előugró oldalon be kell írunk jelszavunkat, hogy érvényesítsük a törlést. A törlés után a honlap kezdőoldalára jutunk vissza.

Ha az ügyfél már a kosarába helyezett termékeket, akkor azokat megtekintheti a kosaram menüpontban, ahol újra átnézheti rendeléseit egy táblázatban összefoglalva. Mikor a vevő kosárba helyez egy terméket az azt eredményezi, hogy az adatbázisban a vásárlásai táblába is megjelenik az ő azonosítója és ahhoz hozzárendelve az áru azonosítója is. Így tudjuk, hogy melyik felhasználó vásárolta az adott azonosítójú terméket. El van tárolva még a darabszám és egy státusz nevű szám, ami ha az áru a kosárba kerül alapértelmezetten egyes értéket vesz fel. A táblázat minden sorának végén található egy törlés gomb, amivel az adott sorban lévő cikket vehetjük ki a kosárból. A táblázatban megjelenik az áru neve, márkája, a választott darabszám és az ár. Az ár oszlop alján pedig láthatjuk, hogy összesen milyen értékben vannak termékek a kosarunkban elhelyezve. Arra is lehetőség van, hogy az összes itt elhelyezett árut egyszerre kivegye a felhasználó. Erre a célra a kosár ürítése feliratú gomb használható. Ha mindent rendben talál és meg kívánja vásárolni az összes a kosárban lévő dolgot a vevő, akkor nincs más dolga, mint megnyomni a vásárlás véglegesítése nevű gombot. Amennyiben a felhasználó adatai nincsenek kitöltve, megjelenik egy felirat miszerint a vásárláshoz ki kell

tölteni az adatait. Ha minden adat megfelelően ki van töltve abban az esetben betöltődik egy új oldal, amin az előbbi táblázat található alatta egy üzenettel. Amiben a felhasználót értesítjük a sikeres vásárlásról és arról, hogy e-mail formájában is megkapta a vásárlást igazoló levelet. Az e-mail a rendelt termékek adatain kívül a felhasználó adatait is tartalmazza, így azokat leellenőrizheti, hogy biztosan jól vannak-e megadva. Ekkor nem csak a vásárló, hanem a cégnél a vásárlásokért felelős személy is kap egy e-mailt, egy titkos másolatot, hogy megkezdhesse a termékek összeállítását postázás céljából. Az adatbázisban is eredményez változást a gomb benyomása mivel a státusza a terméknek innentől kezdve nem kosárban értékű lesz, hanem megvásárolt, tehát a kettes értéket veszi fel.

Megtekinthetőek az eddig a boltban vásárolt termékek is, ez a funkciót a régebbi vásárlásaim menüpontban érhető el. Itt szintén táblázatos formában a vásárlás dátuma alapján sorba rendezve találhatóak a termékek. Az táblázat alsó sorában az eddigi vásárlások összköltsége található. Ez egy fontos érték mivel ez alapján számítható ki a kedvezmény szint, ami az oldal alján tekinthető meg. 50.000 Ft.- alatt nem jár, 50.000 és 100.000 Ft.- között 5% és 100.000 Ft.- vásárlás fölött minden további vásárlásból 10% engedmény jár.

5.2. Adminisztrátori felület

5.2.1. Termékek felvétele

Ha beírom böngészőmbé a <http://localhost/admin.php> címet előugrik egy hitelesítési ablak melyben egy felhasználónevet és egy jelszót kell megadnunk. Amennyiben nem érvényes a kettő közül valamelyik a belépés meg lesz tagadva. Ha azonban ismerjük, a belépéshez szükséges adatokat a következő oldalt érhetjük el.



The screenshot shows the 'Isonzó Papírbolt' admin interface. At the top, there is a header with a paperclip image and the text 'Isonzó Papírbolt' and '--- ADMIN FELÜLET ---'. Below the header, the date 'Március 30, 2011' is displayed. On the left side, there is a vertical menu with the following items: 'Termék Felvétele', 'Termék Törlése/Módosítása', 'Felhasználók törlése', 'Képek feltöltése', 'Rendelések', 'Email lista karbantarása', and 'Email küldése'. The main content area is titled 'Termék felvétele' and contains the following information:

A kategóriák megadása:
1 : Irodai kellékek
2 : Ajándékok
3 : Nyomtatványok
4 : Papír- írószer

Below this, there are four input fields for the product details:

Árú neve:
Ár:
Kategória:
Leírás:
Márka:

At the bottom of the form, there is a button labeled 'Árú hozzáadása'.

Lényegében az oldal felépítése megegyezik a felhasználók által hozzáférhető oldaléhoz, viszont itt a menüpontok a karbantartást szolgálják. Az oldalhoz a menüpontok illetve az oldal felülre eső része szintén külön fájlban van eltárolva, ugyanabból a megfontolásból, mint a felhasználók által használható felületen. Az első menüpont, ami a kezdőoldal is egyben a termék felvétele nevű oldalt teszi elérhetővé. Itt az adminisztrátor feltöltheti az áruk adatait az adatbázisba. Kötelező megadni a termék nevét, árát és kategóriáját, a márka és leírás megadása csak opcionális, de célszerű mivel ezek a felhasználót segítik a termék

kiválasztásában. A kategóriát számmal egytől négyig lehet beállítani. A kategóriáknak megfelelő számok az oldal felsőbb részén fel vannak tüntetve. Ha megfelelően kitöltöttük a mezőket az áru hozzáadása gomb megnyomásával a termék eltárolódik. Valamint az oldal felső részén megjelenik egy kiírás az utoljára felvett termék nevével és kategóriájával.

5.2.2. Termék törlése/módosítása

A következő a sorban a termék törlése/módosítása menüpont, itt táblázatos formában minden az adatbázisban megjelenő termék listázódik csoport szerint azon belül pedig ábécé sorrend alapján. A sorok végén van egy törlés link, amire rákattintva megjelenik a kérdés, hogy valóban törölni szeretnénk-e a terméket. Ha igennel válaszolunk, a termék törlődik az adatbázisból és mi újra a termékek törlése oldalt látjuk magunk előtt, immáron az előbb törölt termék nélkül. Valamint a törlés link mellett van egy módosítás link is, amire kattintva egy adatlap jelenik meg, ahova a már adatbázisban tárolt adatok be vannak írva. Módosíthatjuk a termékek nevét, kategóriáját, árát, leírását és a márkáját is. Ha a kívánt adatokat módosítottuk a módosítás gombbal el is menthetjük a változásokat, vagy a mégse gombbal visszatérhetünk az előző oldalra.

5.2.3. Képfeltöltés

Ebben a menüpontban is az előzőekhez hasonló táblázat fogad minket azzal a különbséggel, hogy itt a törlés helyett a képfeltöltés menüpontot választhatjuk ki. Ha rákattintunk, akkor egy új oldalon láthatunk a termék neve alatt egy tallózás és egy hozzáadás gombot. A tallózásra kattintva kiválaszthatjuk a számunkra megfelelő képet. A kép feltöltésére a PHP beépített \$_FILES függvényét használtam. A kép csak jpg, mpg, gif típusú lehet, ezen kívül a mérete is maximalizálva van 32 kilobájtban. A kép maga a web szerveren tárolódik, ezért kell méretkorlátot megadni, nehogy probléma legyen a helykihasználtsággal. Az adatbázisban csak egy olyan link tárolódik, amely a feltöltött képre mutat, innentől kezdve a link segítségével érhetjük el az általunk kívánt képet. A kép típusát a megjeleníthetőség miatt kell szabályoznunk. A képek felbontása nem lényeges, hiszen a megjelenítésnél fix értékek vannak beállítva így akármennyi pixelpontot tartalmazó kép ugyanolyan méretben jelenik

meg, miközben a felhasználó böngészik az oldalon. Ha kiválasztottuk a számunkra megfelelő képet, akkor csak megnyomjuk a megnyitást. Ezután ha a kép feltöltése gombra kattintunk, akkor, ha valamely érték nem egyezik meg az előre definiáltakkal, akkor hibaüzenetet kapunk, ellenkező esetben megjelenik egy új oldalon, egy üzenet melyben szerepel, hogy a feltöltés sikeres volt. Alatta látható hogy melyik termékhez rendeltük hozzá a képet és maga a kép is megjelenik lentebb. Legalul a vissza a képfeltöltés oldalra linkre kattintva tölthetünk fel más termékekhez is képet.

5.2.4. Felhasználók törlése

Ebben a menüpontban az oldalon regisztrált felhasználók adatait kezelhetjük, amennyiben valamilyen probléma merülne fel velük kapcsolatban. Rendezetten egymás alatti sorokban megjelennek a felhasználók vezeték és keresztnévei illetve e-mail címük. Minden sor elején van egy checkbox amelybe belekattinthatunk, így kijelölve a felhasználókat. Ha kiválasztottunk minden törölni kívánt felhasználót alul a törlés gomb segítségével véglegesíthetjük döntésünket. Ezzel egyidejűleg a felhasználó kap az e-mail címére egy levelet, melyben tájékoztatást kap arról, hogy törölve lett az adatbázisból, és ha ezzel nem ért egyet, akkor milyen elérhetőségen tud reklamációt benyújtani.

5.2.5. E-mail lista karbantartása

Ez a menüpont pontosan úgy néz ki, mint az előző oldal, viszont itt az e-mail listára feliratkozott felhasználók adatait lehet törölni az adatbázisból, szintén egy-egy checkboxok és egy törlés feliratú gomb segítségével.

5.2.6. Rendelések

A rendelések nevű menüpontban találhatóak azon rendelések, amelyek még nem lettek teljesítve. Teljesítés alatt azt értjük, hogy a terméket kézhez is kapta a megrendelő. Táblázatban van felsorolva minden rendelés ahol a teljesített érték még nulla, amint egyesre változik, azaz teljesített státuszú lesz, már nem jelenik meg a táblázatban. A termékek

azonosítója, a felhasználónév, a felhasználó azonosítója és címe, valamint a vásárlás azonosítója jelenik meg soronként. Ha rákattintunk valamely azonosítóra a következő oldalon megtekinthetjük a vásárlás részleteit. A részletekben megjelenik az összes egyszerre megvásárolt termék neve, azonosítója, a megrendelt mennyiség és az hogy nem teljesített a termék státusza. A táblázat alatt található egy Teljesített nevű gomb mely megnyomásával a rendelést teljesített státuszúra állíthatjuk. A gombot megnyomva visszakerülünk a rendelések nevű táblázathoz.

5.2.7. E-mail küldése

Ebben a menüpontban az összes a levelezési listára feliratkozott felhasználónak küldhetünk üzenetet az éppen aktuális akciókról, változásokról, egyéb fontos eseményekről. Az oldalon megjelenik felső részen egy e-mail tárgy mező és alatta egy e-mail szövege mező is melybe az üzenet szövegét gépelhetjük be. Ez egy text típusú mező. Mindkét mező kitöltése kötelező ahhoz, hogy elküldhessük az üzenetet. Ha ezt nem tesszük meg hibaüzenetet kapunk eredményül. Amennyiben minden rendben ki van töltve, a mehet gombra kattintva elküldődik az e-mail és kapunk egy visszajelzést, amiben az összes címet olvashatjuk ahova el lett küldve az általunk megírt szöveg. A levelezésre az PHPmailer függvénykönyvtárat használtam ebben az esetben is. Ezért előfordulhat, hogy egy angol nyelvű hibaüzenet jelenik meg. Ez azért lehetséges, mert a PHPmailer vizsgál olyan lehetséges hibákat is, amiket én nem tettem meg, és e hibák a PHPmailer-ben előre definiált hibaüzeneteket eredményeznek. Ilyen hibát eredményez, ha az e-mail cím nem megfelelő formátumban lett megadva, ez persze ha minden megfelelően működik nem lehetséges mivel már az email cím formátuma letárolásnál ellenőrzésre kerül. Nem csak azon felhasználókkal tarthatjuk a kapcsolatot, akik a levelezési listára vannak feliratkozva, hanem a vásárlóként regisztrált tagokkal is. Ha a vásárláshoz beregisztrált tagoknak szeretnénk levelet küldeni, akkor az oldal felső részén elhelyezkedő checkbox bepipálásával tehetjük ezt meg. Ez a funkció hasznos lehet, ha tájékoztatni szeretnénk vásárlóinkat például arról, hogy az oldal technikai okok miatt nem lesz elérhető egy bizonyos intervallumban.

5.3. Biztonsági és technikai megoldások

5.3.1. Lekérdezések

Az adatbázis kezelés alapja a lekérdezés. Az adatbázissal való kommunikációhoz is vannak beépített függvények a PHP-ben. Ezek közül hármat emelnék ki, mely minden lekérdezésben előfordul:

- `mysqli_connect()` : Ez a függvény az adatbázishoz kapcsolódik a () között megadott paraméterek pedig sorban a következők: a MySQL kiszolgáló helye vagy IP címe, az adatbázisban használt felhasználónév, az adatbázis jelszava és végül az adatbázis neve.
- `mysqli_query()` : Ezzel a függvénnyel egy lekérdezést intézhetünk egy MySQL adatbázishoz. Ez általában adatok bevitelét vagy éppen kinyerését jelenti.
- `mysqli_close()` : Ez a függvény pedig lezárja a kapcsolatot az adatbázissal.

Felfedezhetjük a hasonlóságot a függvények elején mindig ott áll `mysqli` kifejezés. A régebbi verziókban ez csak `mysql` volt, ami azt jelenti, hogy a MySQL-el szeretnénk kommunikálni. Az *i* betű a végén az *improved* vagyis javított, tovább fejlesztett rövidítése, ma már ezen függvények használata ajánlott.

A táblák létrehozása, feltöltése és módosítása a következő parancsokkal lehetséges:

- `CREATE DATABASE adatbázis_neve;` : létrehozhatunk egy adatbázist
- `CREATE TABLE táblanév(
Mező1 típus [feltétel]
Mező2 típus [feltétel]
....
[táblafeltételek]);` : Egy tábla és annak mezőinek létrehozása.
- `ALTER TABLE táblanév utasítások;` : A meglévő tábla módosítása
- `INSERT INTO tábla (oszlop1,oszlop2...) VALUES(érték1,érték2...);` : Adatok beszúrása táblába. Esetleg adhatunk meg feltételeket is a `WHERE` parancs segítségével.

- UPDATE tábla SET oszlop = érték WHERE feltétel; Az adatok módosítása a táblában, de csak ahol a feltételeknek megfelelő érték áll.
- DELETE FROM tábla WHERE feltétel; : Törlés onnan ahol a feltétel igaz.

Adatok lekérdezése táblából:

Az adatok lekérdezésére a SELECT parancs szolgál. A SELECT * FROM tábla1 parancs a tábla1 összes elemét hozza eredményül. Ám hogy a SELECT parancs igazán hasznos legyen számunkra nem csak azt állíthatom be hogy honnan kérdezzek le adatokat, hanem beállíthatok feltételeket is. Ezt a WHERE parancsszó után tehetem meg. Valamint az eredménylistámat rendezhetem is a számomra legmegfelelőbb módon a ORDER BY paranccsal:

ORDER BY oszlopnév RENDEZÉS.

A rendezés két féle lehet:

- ASC : növekvő sorrendben (alapértelmezett)
- DESC : csökkenő sorrendben

A fentebb említett összes parancsot számtalanszor legépeltem mire végül összeállt az alkalmazásom. Sokszor ezen alap lekérdezések azonban nem bizonyultak megfelelőnek az adott helyzetben. Ekkor a megoldást az összekapcsolás, azaz a join jelentette. A tábla összekapcsolásnak több fajtája is létezik, de a legnépszerűbb és általam is használt a belső összekapcsolás (inner join). Ezzel két táblából választhatunk ki sorokat egy feltétel alapján.

```
SELECT tábla1.oszlopnév, tábla2.oszlopnév
FROM tábla1
INNER JOIN tábla2
ON (tábla1.oszlopnév=tábla2.oszlopnév) vagy USING (oszlopnév)
WHERE feltétel
```

Jól látszik, hogy a tábla1 és tábla2-ből vesszük az adatokat, pontokkal összekötve az oszlopnevek vannak felsorolva, mivel több táblából származnak az adatok pontosan meg kell adnunk, hogy az adott oszlop hol található. Az ON sor helyett írhatjuk az USING parancsot is ha a két oszlopnév megegyezik egymással.

5.3.2. A kereső

Az ötödik fejezetben már említett kereső működése csupán csak annyiban tér el egy egyszerű lekérdezéstől, hogy a WHERE feltétel után nem egy általunk megadott feltétel található, hanem a felhasználó által begépelte szó, szavak kapnak helyet, így tulajdonképpen a felhasználó feltételeivel tudunk keresni. Keresni csak a termékek nevére lehet. Ha beír valamit a keresőbe a felhasználó és megnyomja keresés feliratú gombot, akkor a keresendő szót az \$user_search változóban tárolva adjuk át a search.php fájlnek. Ahol lefut a következő lekérdezés:

```
SELECT * FROM bolt_termekek WHERE t_nev LIKE '%$user_search%
```

Az egyenlőségjel helyett a LIKE feltétel van használatban, mivel ha az egyenlőség jelet használtam volna, akkor igen csak kevés találatot eredményez a keresés. A LIKE-nak köszönhetően nem csak a pontosan egyező szavak jelennek meg hanem, olyan találat is szerepel, amely nem pontos egyezést mutat. Valamint a \$user_search változó elé és mögé került a % jel, ezek helyettesítő karakterek. Az SQL-ben a százalékjel tetszőleges, akár nulla karaktert is helyettesíthet. Így a felhasználó által megadott feltételnek nem kell pontosnak lennie elég ha szerepel egy része a kifejezésnek a termék nevében.

5.3.3. PHPmailer függvénykönyvtár

A PHP beépített mail() függvénye segítségével egyszerű szöveges, vagy pedig egy egyszerű HTML levelet tudunk küldeni. Lehetőségünk van UNIX/Linux környezetben a helyi sendmail() paranccsal, Windows alatt pedig SMTP szerver segítségével elküldenünk a levelet. Ezzel a megoldással az a nagy probléma, hogy nem támogatja a hitelesítést ezért döntöttem a PHPmailer mellett.

Egy rövid példa használatára:

```
32 require_once("PHPMailer/class.phpmailer.php");
33 $mail = new PHPMailer();
34 $mail->IsSMTP();
35 $mail->Mailer = "smtp";
36 $mail->SMTPAuth = true;
37 $mail->SMTPSecure = "ssl";
38 $mail->Port = 465;
39 $mail->Username = "e-mail cím";
40 $mail->Password = "jelszó";
41 $mail->Host = "smtp.gmail.com";
42 $mail->From = "küldő e-mail címe";
43
44 $mail->AddAddress($to);
45
46 $mail->Subject = $subject;
47 $mail->Body = $msg;
48 $mail->WordWrap = 50;
49
50 if(!$mail->Send())
51 {
52     echo "Hiba üzenetküldés közben";
53     echo "Mailer Error: " . $mail->ErrorInfo;
54 }
55 else
56 {
57     echo "Üzenet elküldve: ".$to."</br>";
58 }
59
```

A PHPmailer egy mindenki számára ingyenesen hozzáférhető függvénykönyvtár. Példámban az első sor az osztály meghívására szolgál. Ezután egy egyedet készítünk a PHPmailer() osztályból. A következő sorokban beállítjuk a paramétereit a levélnek. Ez elég egyértelmű mivel a PHPmailer osztály nagyon beszédes neveket használ. A From a levél feladóját határozza meg, a Subject a levél tárgyát, míg a body a levél tartalmát. Az SMTPAuth-ban beállítható igényel-e hitelesítést a kiszolgálónk, az SMTPSecure-ben pedig ennek fajtáját állíthatjuk be. A Port sorban a portnak a számát kell beállítani míg a Host sorban a kiszolgáló elérhetőségét, ez nálam a smtp.gmail.com mivel egy @gmail.com végződésű e-mail címről bonyolítom a levelezést. Végül elküldjük a levelet a Send eljárás meghívásával. Ha ennek a visszatérési értéke hamis, akkor hiba történt az elküldés folyamán, a hiba okát az ErrorInfo tartalmának kiírásával jelenítjük meg. Ha a visszatérési érték igaz, akkor az üzenet elküldve megerősítő üzenetet kapjuk eredményül. Ha ezt a kódrészletet belehelyezzük egy while vagy egy for ciklusba akkor akár több e-mail is elküldhető egymás után, csak arra kell oda figyelni hogy a meghívásnál ne a require() függvényt használjuk hanem a require_once() függvényt, mert az előbbi azt eredményezi, hogy csak az első címzett fog levelet kapni tőlünk.

5.3.4. Adatok átadása

A `$_POST` egy különleges változó, amit szuperglobálisnak hívnak, mert a PHP beépített függvénye és a parancsfájl minden részéről elérhető. A `$_POST` már létezik, amikor a parancsfájl lefut, tehát nem kell létrehozni, mint a többi PHP változót. Ez egy tömb, amiben az űrlapok között átadandó adatokat tárolhatjuk. Például:

```
<form method = „post” action = „cel.php”>
<label for=„vez.nev”>Vezetéknév:</label>
<input type=„text” id=„vez.nev”name=„vez.nev”/>
</form>
```

Ekkor form elején meghatároztuk, hogy a vez.nev nevű mező tartalmát a `$_POST` változóban tároljuk, és átadjuk azt a cel.php fájlban, ahol újra felhasználhatjuk azt. Erre a következő módon lehet hivatkozni már a cel.php fájlban belül: `$vezeteknev = $_POST['vez.nev'];`. Ekkor a régebbi vez.nev helyett már az új parancsfájlban és új néven `$vezeteknev` néven elérhető az adat. Az adatok átadására létezik még egy nagyon jó módszer a GET-kérelem. A használata hasonló a post-hoz. Viszont a GET használható arra is, hogy a URL-be paraméterként becsomagolt adatokhoz hozzáférjünk. A GET és POST közötti valódi különbséget a kérelem célja jelenti. A GET-et arra használjuk, hogy adatokat olvassunk ki a kiszolgálóról, anélkül, hogy ott bármire hatással lennénk. Ezzel szemben a POST jellegzetesen arra használatos, hogy adatokat küldjön a kiszolgálónak, ami után a kiszolgáló állapota valamilyen módon megváltozik a kapott adatok függvényében.

5.3.5. CAPTCHA

Az alkalmazásban több célból is regisztrálhat a felhasználó, napjainkban sajnálatos módon sok olyan robot fellelhető az interneten, amelyeknek csak egy céljuk van megszerezni minél több e-mail címet vagy saját reklámokkal, spamekkel elárasztani oldalainkat. Tehát valahogyan meg kell különböztetnünk egymástól az igazi vásárolni kívánó embereket és a robotokat. Erre kiváló megoldást nyújt számunkra egy CAPTCHA program. A mozaikszó jelentése: **C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part azaz magyarul teljesen automatizált nyilvános Turing-teszt számítógépek és emberek megkülönböztetésére. Minden ilyen teszt arra épül, hogy kérdez valami olyan dolgot amelyet

egy ember egyszerűen meg tud válaszolni ám egy reklám robotnak már jóval nagyobb kihívást jelent. Ebben az alkalmazásban ez a teszt egy hat karakterből álló betűsort jelent. Ezt azonban még könnyedén felismerhetné egy optikai karakterfelismerésre programozott robot, így hogy ellehetetlenítsük a szöveget még véletlenszerű pontok és vonalak takarják oly módon, hogy egy ember számára ne jelentsen problémát a felismerése. Példa egy egyszerű CAPTCHA képre:

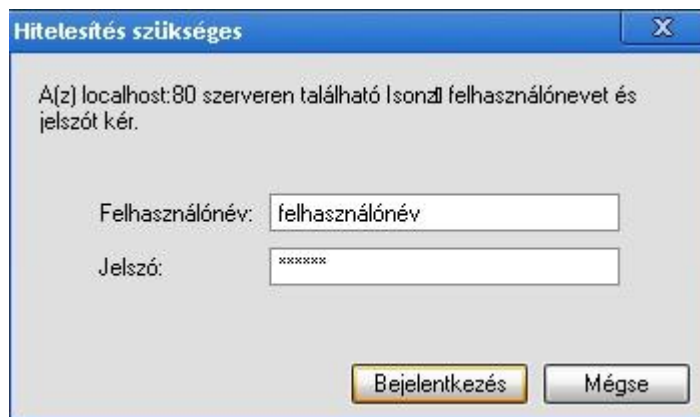


A PHP két beépített függvénye nyújt segítséget a hat karakter előállításához a `rand()` és a `chr()`. A `rand()` egy véletlen számot generál míg a `chr()` függvény a számszerű ASCII karakterkódot alakítja át tényleges karakterré. Jelen esetben a kis betűkre van szükségünk, melyek a 97 és a 127-es tartományba esnek. Ha már előállítottuk a betűsorozatot csak el kell rejtenünk a robotok elől néhány vonallal és ponttal ebben a GD grafikai könyvtárban fellelhető függvények lesznek segítségünkre. Először létrehozunk egy üres képet megfelelő mérettel `imagecreatetruecolor(szélesség, magasság)` függvénnyel. Majd meg kell adnunk a használni kívánt színeket a `imagecolorallocate()` függvénnyel. A pontokat az `imagepixel()`, míg a vonalakat az `imageline()` segítségével rajzolhatjuk ki. Majd a már elkészített szöveget is belerajzoljuk a CAPTCHA képbe és `.png` kiterjesztéssel átadjuk a képet a böngészőnek megjelenítésre. Minden futtatáskor egy ilyen képet kapunk eredményül csak más megjelenített betűkkel.

Most már csak meg kell jeleníteni a képet egy szövegmező mellett ahova a felhasználó begépelheti az általa látott karaktereket. Mivel a `captcha.php` minden alkalommal mikor meghívjuk, egy új `.png` kiterjesztésű képet eredményez, ezért egyszerűen az `` segítségével meg is jeleníthetjük, mint bármely más `.png` kiterjesztésű képet. Ezek után csak össze kell hasonlítani a felhasználó által beírt és az eredeti szöveget, így könnyedén eldönthető, hogy egy valós emberrel állunk-e szemben.

5.3.6. HTTP-hitelesítés

Az alkalmazás rendelkezik admin oldalakkal is melyekhez nem szabad illetékteleneknek hozzáférnie. Erre egy egyszerű és jó megoldást kínál a HTTP-hitelesítés. Ezt a módszert alkalmazva mikor beírjuk a böngészőnkbe, hogy <http://localhost/admin.php> még mielőtt akármit tehetnénk, felugrik egy ablak mely egy felhasználó nevet és egy jelszót kér tőlünk.



A HTTP hitelesítési (authentication) funkciók csak akkor elérhetőek PHP-ben, ha az Apache modulként fut. Az Apache modulként futó PHP esetében a `header()` függvényt kell használni arra, hogy egy "Authentication Required" üzenetet küldjön a kliens böngészőnek, aminek hatására az egy felhasználónév/jelszó bemeneti ablakot nyit meg a felhasználó számára. Ha a látogató kitöltötte a mezőket, az URL, ami a PHP szkriptre mutat, ismét meghívásra kerül, és rendelkezésre állnak a `$PHP_AUTH_USER`, `$PHP_AUTH_PW` és a `$PHP_AUTH_TYPE` változók, amik a felhasználói név, jelszó és azonosítási típus értékeket tartalmazzák értelemszerűen. Mivel az admin felület több oldalból áll mindegyiket meg kell védeni az illetéktelenektől így nem elég csak az `admin.php` oldalt hitelesítéssel ellátni. Létre hoztam a web szerveren egy `hiteles.php` fájlt, amit minden admin felületű oldal elején meghívásra került, így akárhova próbálunk belépni mindig meg kell adni az érvényes felhasználónevet és jelszót. Ez a munkát nem nehezíti meg mivel az összes oldal egy tartományba esik, így nem kell külön-külön minden oldal megnyitásakor megadni a jelszót, elég csak a legelső alkalommal.

5.3.7. Az SHA() függvény

A bejelentkezéshez minden felhasználónak saját jelszava és neve van. A jelszavakat nem célszerű mindenki számára olvasható formában tárolni. Itt alkalmazhatjuk a MySQL SHA() nevű függvényét. Amely az SHA titkosítási algoritmust használja egy szöveges karakterláncra. A rövidítés jelentése Secure Hash Algorithm. A kapott eredmény egy negyven karakter hosszúságú tizenhatos számrendszerű karakterekből álló sztring lesz. Ezt tároljuk az adatbázisban az userpass táblában. Az eredeti jelszó hosszától nem függ, a titkosított jelszó hossza az mindig negyven karakter lesz. Az SHA titkosítás azért is biztonságos, mert az eredeti jelszó nem fejthető vissza a titkosított jelszóból. Tehát ha valaki hozzá is fér a titkos jelszóhoz nem fogja tudni visszafejteni az eredeti jelszót. A beléptetésnél így a jelszó érvényességének ellenőrzéséhez, a felhasználó által begépet jelszót minden alkalommal át kell alakítanunk az SHA() függvénnyel hogy össze tudjuk hasonlítani az általunk tárolt jelszóval. Ez könnyedén megoldható, mert az SHA() függvény ugyanarra a szóra akárhányszor ismételve ugyanazt a titkosított karakterláncot adja eredményül.

5.3.8. A süтик

A süтик vagy angolul cookies használata egy megfelelő módszer arra, hogy a már bejelentkezett felhasználók adatait maradandóan tárolhassuk. A HTTP- hitelesítéssel is megoldható az adatok tárolása ám az nem támogatja a kijelentkezést. A süтик nagyon hasonlítanak a PHP változóira azzal a lényeges különbséggel, hogy a böngésző bezárása vagy a számítógép kikapcsolása után is megmaradnak. Ami fontos még, hogy a süтик törölhetőek is, tehát ha már nincs rájuk szükség, eltávolíthatjuk őket. Ez nagyon jól kihasználható, például ha a felhasználó ki szeretne jelentkezni. A süтик adatait a böngésző tárolja el a felhasználó számítógépén. Az itt tárolt adatokhoz a PHP kódból hozzáférhetőek. A süтик tehát adatokat tárolnak egyedi nevek alatt. Nagy előnyük, hogy beállíthatunk lejáratási időt is, ami azt az időpontot jelöli, amikor a süti automatikusan megsemmisül, ha nincs lejáratási idő beállítva úgy viselkedik mint egy PHP változó, akkor semmisül meg mikor bezárul a böngésző. Ebben az alkalmazásban azért van szükség a sütikre mert sok oldalon keresztül kell emlékezni az adott

felhasználó nevére, azonosítójára, jelszavára és e-mail címére. Így nem kell mindig ezekhez az adatokhoz külön lekérdezést írni, hanem a `$_COOKIE` szuperglobálisban a nevükkel hivatkozhatunk rájuk. Egy süti értékét beállítani a `setcookie(' süti neve', ' tárolandó értéke')` nevű függvénnyel lehet. Megadható egy harmadik érték is a lejáratási idő, ám ez nem kötelező. A sütiben az adatok mindig szöveggént tárolódnak. Amikor a felhasználó bejelentkezik a `login.php` oldalon az előbb említett adatok eltárolódnak a sütiben. Bizonyos menüpontok csak akkor elérhetőek, ha felhasználó már bejelentkezett, ekkor egyszerűen megvizsgáljuk, hogy a felhasználó be van-e léptetve. Ezt az `isset($_COOKIE(' username '))` függvénnyel gyorsan és egyszerűen megtehetjük, ahelyett, hogy többsoros lekérdezésekkel kellene terhelnünk a PHP szerveret. Ha a felhasználó ki szeretne jelentkezni csak a kijelentkezés menüpontra kell kattintania. Ahhoz hogy, ekkor ne maradjanak meg a sütiben beállított értékek törölnünk kell azokat. Ezt úgy érhetjük el ha a süti lejáratási idejét egy múltbeli időpontra állítjuk be. Tehát negatív értéket írunk a lejáratási időhöz.

5.3.9. A szabályos kifejezések vizsgálata

Az oldalon sok mező található ahol a felhasználók adatokat vihetnek fel, mind a felhasználói mind az admin részekben. Ennek a biztosítására, hogy a mezők ne legyenek véletlenül helytelenül kitöltve, és ezzel az adatok tárolását, használatát, valamint a lekérdezések eredményeit megzavarják bizonyos formai követelményeket kellett beállítani. Ezek a beállítások nagyon fontosak, hiszen jól átgondoltnak és megírtak kell lenniük ne legyenek véletlenül olyan adatok is hibásnak találjanak, amelyeknek megfelelőnek kellene minősülniük. A karakterláncok vizsgálata a PHP kiváló módszert kínál, szabályokat hozhatunk létre, ami lehetővé teszi, hogy mintákat keressünk a karakterláncokban. Ezeket a szabályokat hívjuk szabályos kifejezéseknek, vagy angolul `regular expression`-nak röviden `regex`-nek. A szabályokat a meta karakterekből építhetjük fel ezek közül a leggyakoribbak:

- `\d` Ez a meta karakter egy számjegyet keres 0-9 bármely karakter illeszkedik rá. Ha több karaktert szeretnénk keresni, akkor azt vagy a `\d` többszöri leírásával, vagy pedig a `\d` után írt `{}` jelek közé beírunk egy számot akkor annyi karaktert fogunk keresni amekkora szám ott szerepel.

- \ w Egy tetszőleges alfanumerikus karaktert kereshetünk vele, vagyis számot 0-9 között, vagy betűt a-z és A-Z között.
- \ s Ez a karakter egy üreshely-karaktereket (whitespace) keres. Üreshely-karakter lehet a szóköz, az újsor, a tabulátor is.
- ^ Ez a jel egy karakterlánc kezdetét keresi, tehát azt jelzi a szabályon belül, hogy az illeszkedés a karakterlánc elején kell, hogy kezdődjön, nem pedig akárhol máshol.
- \$ Ez a meta karakter pedig a karakterlánc végét keresi. Tehát a ^ és a \$ jelek segítségével pontosan behatárolhatjuk egy karakterlánc hol kezdődjön és hol végződjön.
- • A pont karakter bármilyen karakterre illeszkedik az újsor kivételével.

Ezekon a meta karaktereken kívül alkalmazásomban használtam egy fontos függvényt amely nélkül igen nagy kihívás lett volna az érvényes e-mail címeket ellenőrizni. Ez a `checkdnsrr()` függvény, amely nem csak azt vizsgálja meg, hogy az adott karakterlánc elfogadható-e levelezési tartományként, hanem ellenőrzi a DNS-bejegyzések között, hogy a tartománynév létezik-e. Az elkészített szabályos kifejezésekét pedig a `preg_match()` függvény segítségével lehet összehasonlítani a karakterláncokkal. Ez a függvény eredményül nullát ad vissza, ha a karakterlánc eltér a mintától és egyest, ha illeszkedik a mintára. Ezen meta karakterek és függvények segítségével bármilyen illesztési mintát létre lehet hozni, amelyek kiválóan segítik munkánkat és az adatok ellenőrzését.

6. Összefoglalás

Dolgozatomban rálátást nyerhettünk milyen eszközökkel és programozási nyelvekkel lehet létrehozni egy dinamikus weboldalt. Észrevehettük, hogy egy weboldalon, még ha az külsőleg egészen egyszerű felépítésű, a háttérben rengeteg a felhasználó által nem látott funkció és lekérdezés futhat. Megismerkedhettünk a PHP nyelv alapjaival, és láthattuk egyszerűségét, használhatóságát, azt, hogy miért is olyan népszerű és elterjedt világszerte. Valamint azt is láthattuk, hogy milyen jól használható együtt a MySQL és Apache párossal.

A funkciókat jól elkülönítve mutattam be, azért mert ezeket a részeket célszerű eleve úgy megírni, hogy azok önmagukban újrafelhasználhatóak legyenek. Így a későbbiekben hasznosíthatóak lesznek más alkalmazások fejlesztése során, ily módon gyorsítva a fejlesztő munkáját. A bemutatás érthető és világos betekintést engedett a programírás világába a nem hozzáértők számára is. Az alkalmazást ajánlanám minden olyan kisebb méretű cég számára, aki belevágna a mai világban olyannyira divatos és felkapott internetes értékesítésbe. Az, hogy milyen jellegű kereskedelemmel foglalkozik teljesen lényegtelen, mert az értékesítői oldalak nagyjából ugyanarra a sémára épülnek. Tehát kis módosításokkal átalakíthatóak az alkalmazás részei bármilyen termékek forgalmazására. Ideális esetben módosítani csak a kategóriákat és az adatbázis termékeket tartalmazó részét kell.

A fejlesztés során rengeteg problémával szembesültem, amik között volt olyan is, amelyre végül nem tudtam megoldást találni. Ilyen például az oldal kinézete, amiről első ránézésre látható, hogy nem egy profi ezzel foglalkozó csapat építette fel. Funkcióját maximálisan kiszolgálja, de külleme elmarad az általam eredetileg elvárttól. Ez a flash animációk és a grafikai tudásom hiánya miatt van. A mai honlapok szinte mindegyikén feltűnő technológia rendkívül látványos és szép megoldásokat kínál a honlap kinézetének alakítására. Későbbiekben szándékozom megtanulni a flash programozás alapjait és az eredeti elképzeléseimnek megfelelően kialakítani az alkalmazás kinézetét.

Az internetes kereskedelem fejlődése magyarországi megjelenése óta rendületlenül növekvő tendenciát mutat. Sikerét annak köszönheti, hogy az emberek szeretik a kényelmet tehát

bármely napszakban, bármely napon megrendelhetik a kiszemelt árucikket anélkül, hogy el kellene hagyniuk otthonaikat. Valamint egyre nagyobb a bizalom az online és előre átutalásos fizetésekkel szemben. Az interneten minden megtalálható, amit szeretnénk elérni, szórakozás, játék, információ és bármilyen jellegű szolgáltatás vagy áru.

7. Irodalomjegyzék

- Lynn Beighley, Michael Morrison - Agyhullám: PHP& MySQL
Kiskapu Kft., Debrecen, 2009
- Tim Converse, Joyce Park, Clark Morgan -PHP5 and MySQL® Bible
Wiley Publishing, Inc., Indianapolis, Indiana, 2004
- Virginia DeBolt – HTML és CSS Webszerkesztés stílusosan
Kiskapu Kft., Debrecen, 2005
- http://humanvalue.eu/humanvalues/hu/online/Online_piaci_korkep_2010_Vojnits.pdf
- <http://desingrengeteg.honlapepito.hu/?modul=oldal&tartalom=595029>
- <http://www.epab.bme.hu/projektek/HPMobileGrant/F04-HTML/index.html>
- <http://www.kando-kkt.sulinet.hu/mirror/phphu/intro-history.html>
- <http://ceginfo.computerworld.hu/hirek/ifua/mysql-a-legnepszerubb-nyilt-forraskodu-adatbazis-kezelo-rendszer>
- <http://hu.wikipedia.org/wiki/MySQL>
- <http://hu.wikipedia.org/wiki/CSS>
- <http://hu.wikipedia.org/wiki/JavaScript>

Köszönetnyilvánítás

Szeretném ez úton hálás köszönetemet kifejezni konzulensemnek, Dr. Kuki Attila tanár úrnak a szakdolgozatban és a főiskolai évek alatt nyújtott rengeteg segítséget.

Továbbá köszönöm családomnak, nem csak az anyagi, hanem a rengeteg lelki támogatást is.