

Oskari Väistö

VERKKOKAUPAN SUUNNITTELU JA TOTEUTUS

Liiketalouden koulutusohjelma

Tietojenkäsittely

2012

VERKKOKAUPAN SUUNNITTELU JA TOTEUTUS

Väistö, Oskari
Satakunnan ammattikorkeakoulu
Liiketalouden koulutusohjelma
Huhtikuu 2012
Ohjaaja: Stenfors, Juha
Sivumäärä: 49
Liitteitä:

Asiasanat: PHP, MySQL, verkkokauppa, tietokanta

Opinnäytetyön aiheena oli verkkokauppasovelluksen suunnittelu ja toteutus. Tarkoituksena oli luoda toimiva ja helppokäyttöinen verkkokauppasovellus, jossa hyödynnettäisiin tietokantojen tuomia mahdollisuuksia. Työ toteutettiin ilman asiakasyritystä.

Työn toteuttamisessa käytettiin ilmaista avoimeen lähdekoodiin perustuvaa XAMPP-ohjelmistoa, joka on suunniteltu apuvälineeksi sivustojen suunnittelijoille ja ohjelmoijille. Näin mahdollistettiin esimerkiksi PHP:n ja MySQL:n käytön suoraan omalta koneelta ilman tarvetta erillisen palvelimen asennukselle ja konfiguroinnille. Sivujen rakenne toteutettiin (X)HTML5- ja toiminnallisuus PHP -ohjelmointikielellä. Sivuston ulkoasu saatiin tyylikkäämmäksi käyttämällä CSS -tyylilomakkeita. Työssä käytetyt tietokantaominaisuudet toteutettiin ilmaisella avoimeen lähdekoodiin perustuvalla MySql -tietokantajärjestelmällä. Sivuston elementtien animointiin käytettiin Javascript-kirjasto jQueryä.

DESIGNING AND IMPLEMENTING AN E-COMMERCE APPLICATION

Väistö, Oskari

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in

May 2012

Supervisor: Stenfors, Juha

Number of pages: 49

Appendices:

Keywords: PHP, MySQL, e-commerce, database

The subject of this thesis was to design and implement an e-commerce application. The aim was to create a functional and easy to use application which would make use of possibilities provided by the use of databases. The thesis was executed independently, without a client organization.

In the execution of the thesis was used a free open source Apache distribution called XAMMP, designed as a tool for web designers and programmers. It enabled the use of PHP and MySQL directly from the users own personal computer without the need to install them on an external server. The structure of the e-commerce application was executed using (X)HTML5 and the functionality using PHP. CSS was used to give the layout a more stylish look. The database functionalities used in this thesis were created with a free open source database management system MySQL.

SISÄLLYS

1	JOHDANTO.....	5
2	OHJELMOINTIKIELET.....	5
2.1	(X)HTML.....	6
2.1.1	HTML5.....	6
2.2	CSS.....	7
2.2.1	CSS3.....	9
2.3	PHP.....	9
2.3.1	Istunnot.....	11
2.3.2	Taulukot.....	11
2.4	JavaScript.....	12
2.4.1	jQuery.....	13
3	OHJELMISTOT.....	13
3.1	XAMPP.....	13
3.2	MySQL.....	14
3.3	phpMyAdmin.....	15
4	TIETOKANTA.....	15
4.1	Yleistä tietokannoista.....	15
4.1.1	Relaatiotietokanta.....	15
4.2	Tietokannan rakenne.....	16
5	RAKENNE JA TOIMINNALLISUUS.....	20
5.1	Rakenne.....	20
5.2	Toiminnallisuus.....	22
5.2.1	Päävalikko.....	22
5.2.2	Haku.....	23
5.2.3	Ostoskori.....	28
5.2.4	Viimeksi katsotut.....	31
5.2.5	Pääsivut.....	35
5.2.6	Tuotesivut.....	36
5.2.7	Rekisteröityminen ja kirjautuminen.....	38
5.2.8	Salasanan palautus.....	39
5.2.9	Kassatoiminnot.....	40
6	TUOTEHALLINTA.....	42
6.1	Tuotteiden lisääminen tietokantaan.....	43
7	YHTEENVETO.....	47
	LÄHTEET.....	49
	LIITTEET	

1 JOHDANTO

Verkkokaupassa asiointi on nykyään jo todella yleistä ja pääosin myös helppoa. Asioimaan pääsee vuorokaudenajasta tai pyhäpäivistä riippumatta juuri silloin kuin itse haluaa ja oikein toteutetussa verkkokaupassa asiointi on nopeaa ja vaivatonta. Nykyaikaisessa verkkokaupassa asioidessaan asiakas on tietoinen tuotteiden saatavuudesta ja ostoskorinsa sisällöstä. Hän saa tilauksen tehtyään sähköpostiinsa varmistuksen tilauksen perille menosta, sekä pääsee halutessaan maksamaan tilauksensa suoraan verkkopankkinsa kautta.

Tässä työssä pyrittiin toteuttamaan selkeä ja helppokäyttöinen, mutta samalla myös tarpeeksi näyttävä verkkokauppasivusto. Koska sivuston ei ollut tarkoitus tulla niin sanotusti oikeaan käyttöön, kaikkia hyvälle verkkokaupalle kuuluvia ominaisuuksia kuten esimerkiksi rajapintaa verkkokaupan ja verkkopankkien välille, ei kyetty työssä toteuttamaan. Sivuston sujuvampaan hallintointiin luotiin myös erillinen hallinnointisivu joka helpottaa tuotteiden lisäämistä sivustolle.

Työtä varten luotiin MySQL-tietokanta tarvittavine tauluineen. Tietokantaa hallittiin pääosin PHP -ohjelmointikielellä toteutetulla verkkokauppasovelluksella, johon rakenne toteutettiin XHTML-merkintäkielellä ja ulkoasu CSS-tyylikielellä.

2 OHJELMOINTIKIELET

Verkkokauppasovelluksen perusrakenne toteutettiin XHTML-merkintäkielellä ja ulkoasu muokattiin sopivaksi CSS-tyylilomakkeilla. Sivuston pääasiallinen toiminnallisuus toteutettiin palvelinpuolella pyörivällä PHP-ohjelmointikielellä. Lopuksi sivuston ulkoasuun lisättiin näyttävyyttä animoimalla joitakin elementtejä käyttäen hyväksi JavaScript-kirjasto jQueryä ja sen mahdollistamia visuaalisia efektejä.

2.1 (X)HTML

XHTML (Extensible HyperText Markup Language) eli laajennettava hypertekstin merkintäkieli tarkoittaa XML-kielellä kirjoitettua HTML-dokumenttia, joka muodustuu HTML-kielen komennoista ilman muotoilumäärittelyjä. XHTML-dokumenttiin voidaan myös liittää XML-standardin mukaisia omia elementtejä (Linjama 2001, 9).

HTML-kielessä itse koodiin kirjoitettavat muotoilumäärittelyt toteutetaan XHTML-kielessä erillisin CSS-tyylilomakkein. XHTML-kielessä kaikki elementit, attribuutit ja niiden arvot kirjoitetaan pienaakkosin ja kaikki elementit on päätettävä oikein. Myös tyhjät elementit kuten rivinvaihtoa ilmaiseva `
` on XHTML-kielessä kirjoitettava muotoon `
` (Saarikumpu 1997).

2.1.1 HTML5

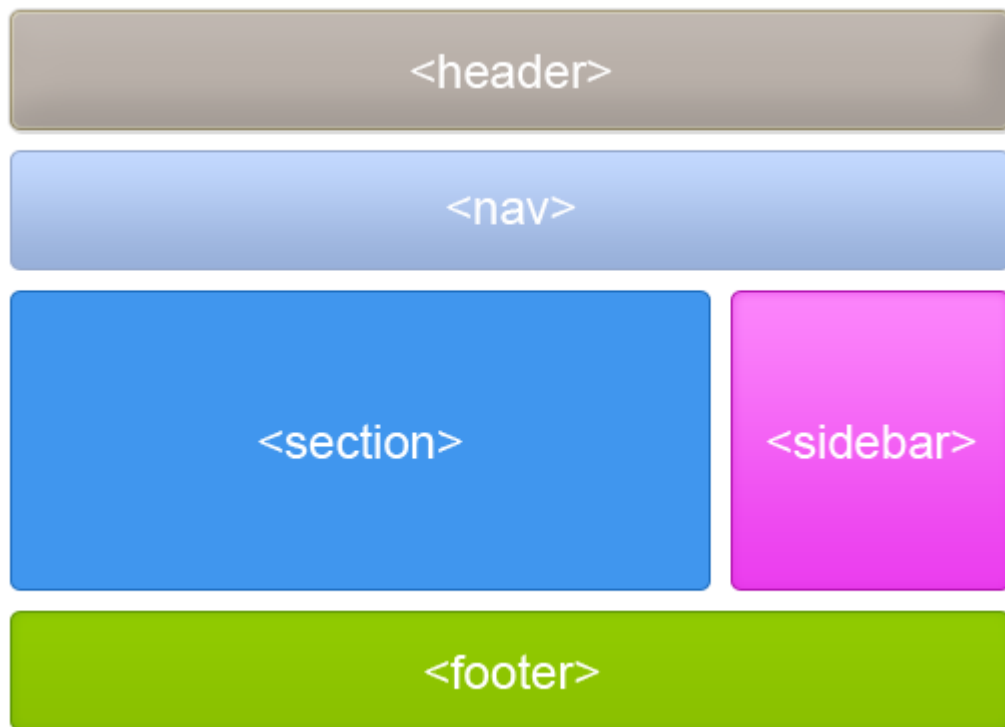
HTML5 on HTML:n uusin järjestelmäriippumaton evoluutio ja tarjoaa monia uusia moderneille verkkosovelluksille tarpeellisia toimintoja. Kaikkien modernien selainten uusimmat versiot tukevat monia HTML5:n ominaisuuksia. HTML5 sisältää monia uusia sivustojen luomista helpottavia ja sivustojen käyttökokemusta parantavia ominaisuuksia ja on suunniteltu taaksepäin yhteensopivaksi olemassa olevien selainten ja vanhempien html-versioiden kanssa. (Sulkko 2011)

Tulevaisuudessa HTML5 mahdollistaa esimerkiksi uusilla video ja audio -tageillaan musiikin tai elokuvien trailereiden esityksen sivuilla ilman mitään kolmannen osapuolen plugineita. Lisäksi sen tuomia ”drag and drop” -ominaisuuksia voi käyttää hyväksi vaikkapa ostoskorisovellusta suunniteltaessa, jolloin asiakas voisi ottaa tuotteesta kiinni ja siirtää sen hiirellä ostoskoriinsa. Tässä työssä HTML5:n mukanaan tuomista uusista ominaisuuksista on käytetty semaattisia elementtejä.

Erilaisia semaattisia elementtejä:

- **article:** määrittelee riippumattoman itsenäisen sisällön, jonka pitäisi olla järkevä ja jaettavissa ilman muuta sisältöäkin.

- **section:** määrittelee dokumenttiin eri osioita kuten luvut, otsikot, alatunnisteet tai muut dokumentin osat.
- **header:** määrittelee dokumentin johdannon tai ryhmän navigaatioelementtejä dokumentille.
- **footer:** määrittelee dokumentin alatunnisteen, joka tyypillisesti sisältää tietoja tekijästä ja dokumentin luontiajankohdasta.
- **aside:** määrittelee asetelukohtansa muusta sisällöstä sivussa olevan, mutta siihen kuitenkin liittyvän sisällön.
- **nav:** määrittelee osion joka sisältää navigaatiolinkit. (Sulkko 2011)



Kuva 1. Sivuston perusrakenne semanttisilla HTML5 elementeillä.

2.2 CSS

CSS (Cascading Style Sheets) on HTML, XHTML ja XML-dokumenttien ulkoasun ja esitystavan määrittelyyn luotu yksinkertainen tyylikieli.

Tyylien etuna on, että käyttäjä voi yhdellä määrittelyn muutoksella vaikuttaa kaikkiin sitä tyyliä käyttäviin elementteihin. Jokaisen elementin määrittely erikseen perinteiseen

tyyliin vaatii jokaisen elementin määrittelyjen muuttamista yksitellen, jotta päästään samaan lopputulokseen. Näin ollen tyylit tarjoavatkin helpon tavan päivittää dokumentin määrittelyjä ja samalla pitää useista dokumenteista koostuvien kokonaisuuksien ulkoasut yhtenäisinä (Schafer 2010, 398).

```

1
2 body {background-color: #000000;}
3
4 p {
5   font-family: verdana;
6   font-size: 12px;
7   color: #ffffff;
8 }
9
10 p.red { color: #ff0000;}
11

```

Esimerkki 1. Esimerkki CSS-tyylimäärittelystä.

Esimerkissä 1 on web-sivun taustaväri asetettu mustaksi (heksadesimaalina #000000) määrittelemällä se body-elementin *background-color* ominaisuudella rivillä 2. Sivun leipätekstielementti (p) on rivillä 5 määritelty käyttämään tekstin fonttina Verdanaa (font-family: verdana) ja rivillä 6 sen kokona 12 pikseliä (font-size: 12px). Tekstin väri on rivillä 7 määritelty valkoiseksi (color: #ffffff). Nämä määrittelyt vaikuttavat siis kaikkiin sivun leipätekstielementteihin. Voidaan kuitenkin erikseen määrätä jotain leipätekstielementtiä poikkeamaan äskeisistä perusmäärittelyistä, kuten esimerkikoodissa on tehty elementin *p.red* kohdalla. Siinä tekstin väri on valkoisen sijaan määritelty rivillä 10 punaiseksi (color: #ff0000). HTML-koodissa tähän poikkeuksellisesti määriteltyyn leipätekstiluokkaan viitattaisiin tagilla: `<p class="red">`. Kaikki CSS-määrittelyrivit päätetään puolipisteeseen.

CSS-tyyli voidaan yhdistää HTML-dokumenttiin kolmella tavalla. Ensimmäinen ja päivityksen kannalta huonoin tapa on kirjoittaa tyylimäärittelyt suoraan jokaiseen elementtiin erikseen esim. `<p style="color: black">Tekstiä</p>`. Toinen ja huomattavasti jo parempi tapa on kirjoittaa esimerkin mukaiset tyylimäärittelyt HTML-dokumentin head-osaan erityisellä style-elementillä. Tämä tapa toimii tosin vain jos tyylimäärittelyt on tehty jokaiseen sivuston dokumenttiin. Kolmas ja ehkä paras tapa onkin liittää HTML-dokumenttiin ulkoinen CSS-tyylitiedosto head-osaan sijoitettavalla

link-elementillä esimerkiksi seuraavasti: `<link rel="stylesheet" type="text/css" href="tyylit.css" />`. Samaan HTML-dokumenttiin voidaan liittää useita tyylitiedostoja.

2.2.1 CSS3

CSS3 on vielä kehitteillä oleva määritelmä. Joitain sen osia pidetään jo yleisesti vakaina ja modernit selaimet tukevat niitä jo hyvin, kun taas jotkut osat ovat vasta kokeiluasteella eikä niille näin ollen löydy vielä tukea kuin vaihtelevasti. Vasta teoreettisten ehdotelmien tasolla oleville ominaisuuksille ei tukea luonnollisestikaan vielä löydy. Kaikesta tästä edellä mainitusta johtuen ennen CSS3:n ominaisuuksien käyttämistä on tärkeää olla tietoinen kunkin ominaisuuden nykyisestä tukiaasteesta (Gasston 2011, 1).

Tässä työssä CSS3:n ominaisuuksia on käytetty joidenkin elementtien kulmien pyöristykseen (`border-radius`) ja varjojen (`box-shadow`) luomiseen. Tämä helpotti ulkoasun vimeistelyä, koska esimerkiksi jokaiselle erikokoiselle kuvalle ei tarvinnut luoda erikseen omia varjoina käytettyjä kuvaelementtejä erillisellä kuvankäsittelyohjelmalla, eikä niiden asemointin CSS:llä tarvinnut käyttää aikaa, vaan koko asia hoitui muutamalla rivillä määrittelyä. Kyseiset ominaisuudet toimivat kaikilla moderneilla internetiselaimilla (IE9, Mozilla Firefox 3.5+, Safari 3+, Opera 10+, sekä uusimmat versiot Google Chromesta) ja vaikka niiden toiminnassa vanhemmilla selaimilla onkin puutteita, ei niiden toimimattomuus kuitenkaan aiheuta kuin lieviä visuaalisia eroja alkuperäiseen tarkoitukseen verrattuna. Katsoin siis hyvin voivani käyttää kyseisiä ominaisuuksia työssäni.

2.3 PHP

PHP (Hypertext Preprocessor) on laajalti käytetty avoimeen lähdekoodiin perustuva ohjelmointikieli, joka soveltuu erityisesti verkkopalvelujen kehittämiseen ja jota voidaan kirjoittaa tavallisen HTML-koodin lomaan (<http://www.php.net>).

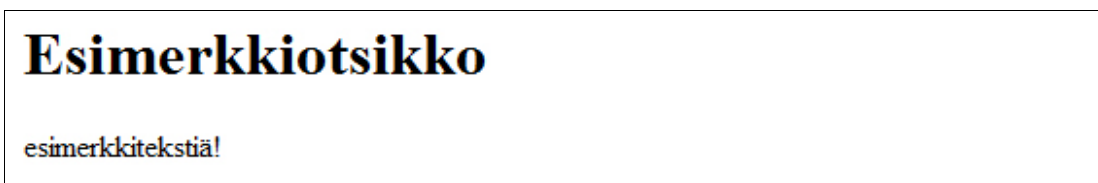
```

1 <html>
2   <head>
3     <title>PHP-esimerkki</title>
4   </head>
5   <body>
6     <h1><?php echo "Esimerkkiotsikko" ?></h1>
7     <?php
8       echo "esimerkkitekstiä!";
9     ?>
10  </body>
11 </html>

```

Esimerkki 2. Esimerkki PHP-koodista.

PHP -koodi aloitetaan aina <?php - ja lopetetaan ?> -tagilla. Esimerkissä (esimerkki 2) on tavallisten HTML-pääotsikkotagien <h1> väliin kirjoitettu pätkä PHP-koodia (rivi 6), jonka tuottama teksti ”Esimerkkiotsikko” tulostuu selaimessa pääotsikkona kuten tagien väliin normaalisti kirjoitettu teksti (Kuva 2). Esimerkin toinen PHP-koodi (rivit 7 - 9) tulostaa näytölle vain tekstin ”esimerkkitekstiä!” ilman mitään muotoiluja. PHP-koodia voidaan siis kirjoittaa joko omina pätkinään tai sitten HTML-tagien väliin ja näin hyödyntää esim. CSS:n tarjoamia muotoilumahdollisuuksia.



Kuva 2. Esimerkkikoodin tuottama www-sivu.

Koska PHP-koodi suoritetaan palvelinpäässä ja vasta sen tuottama HTML-koodi lähetetään asiakkaalle, ei selaimen näyttämästä lähdekoodista (esimerkki 3) pysty päättelemään millä koodilla selaimella näkyvä lähdekoodi on oikeesti tuotettu.

```

1 <html>
2   <head>
3     <title>PHP-esimerkki</title>
4   </head>
5   <body>
6     <h1>Esimerkkiotsikko</h1>
7     esimerkkitekstiä!
8   </body>
9 </html>

```

Esimerkki 3. Selaimen näyttämä lähdekoodi.

2.3.1 Istunnot

Istunnossa (session) itse tiedot tallennetaan palvelimelle ja käyttäjä tunnistetaan hänelle istunnon alkaessa luodulla ja lähetetyllä uniikilla istunto-id:llä. Istunto-id tallennetaan yleensä käyttäjä koneelle evästeenä (cookie), mutta sen kuljetus GET-metodilla on myös mahdollista. Tästä voi ilman eri toimenpiteitä kuitenkin koitua riskinsä tietoturvalle ja kuka tahansa voi päästä käsiksi istunnon tietoihin tietäessään käytettävän osoitteen (wiki.mureakuha.com).

Istunto aloitetaan ja sitä jatketaan aina `session_start()` -funktiolla, joka tarkastaa onko istunto jo olemassa ja tarpeen mukaan jatkaa sitä tai aloittaa sen. Funktio on lisättävä jokaisen sivun alkuun, jolla on tarkoitus päästä käsiksi istuntomuuttujiin.

```
1 <?php
2 session_start();
3 $_SESSION['username'] = "majja.meikalainen";
4 echo $_SESSION['username'];
5 ?>
```

Esimerkki 4. Tietojen tallentaminen ja tietojen tulostus istunnossa.

Esimerkissä 4 avataan istunto rivillä 2 käyttämällä `session_start()` -funktiota. Rivillä 3 tallennetaan istunnon muuttujalle ”username” arvo ”majja.meikalainen”. Rivi 4 tulostaa muuttujan ”username” arvon käyttäjän näytölle.

Tässä työssä sessioita käytetään muun muassa ostoskorin toteutuksessa.

2.3.2 Taulukot

PHP:ssä taulukolla (array) voidaan yhdelle muuttujalle antaa useita eri arvoja. Eri arvot (value) löydetään taulukosta sen sisältämien avainten (key) avulla. Taulukon määrittelyyn on monta tapaa:

```
1 <?php
2 $taulukko = array("Ensimmäinen", "Toinen", "Kolmas");
3 echo $taulukko[1];
4 ?>
```

Esimerkki 5. Taulukun luonti `array()` -funktiolla.

Taulukon luomiseen voidaan käyttää `array()` -funktiota (rivi 2). Esimerkissä 5 on määritetty vain taulukon solujen sisältämät arvot, joten PHP määrittelee soluille avaimet automaattisesti. Ensimmäisen solun avaimeksi tulee näin 0, toisen 1 ja kolmannen 2. Rivillä 3 tulostetaan toisen solun arvo ”Toinen” näytölle.

```

1 <?php
2     $taulukko[] = "Ensimmäinen";
3     $taulukko[] = "Toinen";
4     $taulukko[] = "Kolmas";
5     echo $taulukko[1];
6 ?>

```

Esimerkki 6. Vaihtoehtoinen tapa saman taulukon muodostamiseen.

Esimerkin 5 taulukko voidaan myös luoda ilman `array()` -funktiota esimerkin 6 tavalla. Lopputuloksena on sama taulukko ja tulostus kuin esimerkin 5 tilanteessa.

```

1 <?php
2     $taulukko1 = array("eka" => "Ensimmäinen", "toka" => "Toinen", "kolmas" => "Kolmas");
3
4     $taulukko2["eka"] = "Ensimmäinen";
5     $taulukko2["toka"] = "Toinen";
6     $taulukko2["kolmas"] = "Kolmas";
7     echo $taulukko2["kolmas"];
8 ?>

```

Esimerkki 7. Avainten määrittely manuaalisesti.

Esimerkissä 7 luodaan kaksi identtistä taulukkoa (`$taulukko1` ja `$taulukko2`) antamalla soluille avaimet manuaalisesti. Taulukon ja avainten arvoina voi olla lähes mitä tietotyyppiä tahansa, jopa samassa taulukossa sekaisin. On kuitenkin suositeltavaa käyttää vain kokonaislukuja ja merkkijonoja.

2.4 JavaScript

Modernien selainten yleisesti tukema JavaScript on oliopohjainen niin sanottu scriptikieli, jonka koodia ei erikseen käännetä omaksi ohjelmakseen vaan se tulkitaan sellaisenaan. Toimiakseen JavaScript tarvitsee isäntäympäristön kuten web-selaimen, jonka JavaScript-tulkki suorittaa koodin rivi kerrallaan. Vahvuuksina JavaScriptillä on sen keveys, tehokkuus, järjestelmäriippumattomuus sekä yksinkertainen ja vapaamuotoinen syntaksi (Saarikumpu, O. <http://weppipakki.com>).

Työssä JavaScriptillä toteutettiin tuotesivujen ”nimikkeitä per sivu”- ja järjestysominaisuudet, sekä hakupalkin ominaisuus, jolla hakuruudussa oleva teksti ”Hae Viihdevarastosta” saadaan häviämään kun ruudun valitsee hiiren klikkauksella.

2.4.1 jQuery

jQuery on nopea ja ytimekäs JavaScript-kirjasto, joka yksinkertaistaa HTML-dokumentin tapahtuma-hallintaa, animointia ja yhteistyötä Ajaxin kanssa mahdollistaen nopeamman web-kehityksen (<http://www.jquery.com>).

Tässä työssä jQueryä on käytetty esimerkiksi menun alavetovalikoiden ja rekisteröintipalkin animointiin.

3 OHJELMISTOT

Koska PHP-koodi suoritetaan serveripuolella vadietaan sen suoritukseen ja sivuston testaukseen joko serveri, jolle PHP on asennettu, tai vaihtoehtoisesti joku serveriympäristöä emuloiva ohjelmisto. Tämän työn suorituksessa on käytetty XAMPP-nimistä ohjelmistoa.

3.1 XAMPP

XAMPP (Cross-platform, Apache HTTP Server, MySQL, PHP, Perl) on ohjelmisto joka asentaa Apachen, MySQL:n, PHP:n ja Perl:n käyttäjän koneelle, jonka jälkeen normaalisti erillistä palvelinympäristöä vaativat PHP ja MySQL saadaan toimimaan suoraan käyttäjän kotikoneelta tarvitsematta rakentaa erikseen palvelinympäristöä niille. Koska palvelinympäristön rakennus ei ollut osa työtäni, päätin nopeuttaa työn aloitusta ja helpottaa työn aikaista testausta käyttämällä kyseistä ohjelmistoa.

3.2 MySQL

MySQL on maailman suosituin avoimeen lähdekoodiin perustuva tietokantaohjelmisto. Sen käyttäjinä ovat monet maailman suurimmista ja nopeiten kasvavista organisaatioista mukaanlukien sellaiset yritysmaailman jätit kuten Yahoo!, Alcatel-Lucent, Google, Nokia, Youtube, Wikipedia sekä Booking.com (<http://www.mysql.com>). Tämän työn tietokantaominaisuudet on toteutettu MySQL:llä.

MySQL-tietokantaa pystytään käyttämään sivustolta PHP-kielisin komennoin. Seuraavassa esimerkki yhdistämisestä MySQL-tietokantaan PHP:lla.

```
1 <?php
2
3     $server = "localhost";
4     $user = "testuser";
5     $pass = "testi";
6     $db = "varasto";
7
8     mysql_connect($server, $user, $pass) or die("Ei yhteyttä tietokantaan!");
9
10    mysql_select_db($db) or die("Tietokannan valinta epäonnistui!");
11
12 ?>
```

Esimerkki 8. Yhdistäminen MySQL-tietokantaan.

Esimerkin 8 riveillä 3 – 6 asetetaan muuttujien \$server (serverin osoite), \$user (käyttäjätunnus), \$pass (salasana) ja \$db (tietokannan nimi) arvoiksi vaaditut arvot. Varsinainen yhteys tietokantapalvelimeen muodostetaan mysql_connect() -funktiolla, jonka parametreiksi asetetaan edellä määritellyt muuttujat \$server, \$user ja \$pass (rivi 8). Tällöin ohjelma yrittää muodostaa yhteyden serverille "localhost", käyttäjätunnuksella "testuser" ja salasanalla "testi". Jos annetuin parametrein ei pystytä muodostamaan yhteyttä, tulostaa rivin 8 kohta "or die()" käyttäjälle virheilmoituksen "Ei yhteyttä tietokantaan!". Jos taas yhteyden muodostaminen onnistuu, valitaan tietokanta funktiolla mysql_select_db(). Esimerkkitapauksessa valitaan tietokanta nimeltä varasto (rivi 10). Tämän jälkeen on mahdollista suorittaa kyselyjä valittuun tietokantaan. Virhetilanteessa tulostetaan käyttäjälle ilmoitus "Tietokannan valinta epäonnistui!".

3.3 phpMyAdmin

PhpMyAdmin on ilmainen PHP:llä toteutettu työkalu MySQL-serverin hallintaan netin yli. Se tukee laajalti erilaisia operaatioita joista useimmin käytettyjä (kuten tietokantojen, pöytien, kenttien, relaatioiden, indeksien, käyttäjien ja käyttöoikeuksien hallinta) voi käyttää suoraan graafisella käyttöliittymällä. Vaihtoehtoisesti käyttäjä voi halutessaan suorittaa tavallisia SQL-kielisiä kyselyitä (<http://www.phpmyadmin.net>).

Tämän työn käyttämä tietokanta relaatioineen on tehty XAMPP-ohjelmistopakettin mukana tulleella phpMyAdminilla.

4 TIETOKANTA

4.1 Yleistä tietokannoista

Tietokanta on loogisesti yhteenkuuluvien, tietokantakielellä (kuten SQL) hallintoitujen, tallennettujen tietojen joukko. Tietokannassa olevia tietoja hallinnoidaan erityisellä tietokannan hallintojärjestelmällä, TKHJ (Database Management System, DBMS). Esimerkkejä hallintojärjestelmistä ovat Oracle, DB2, Microsoft SQL Server, MySQL ja Access (Hovi, Huotari, Lahdenmäki, 2005, 4).

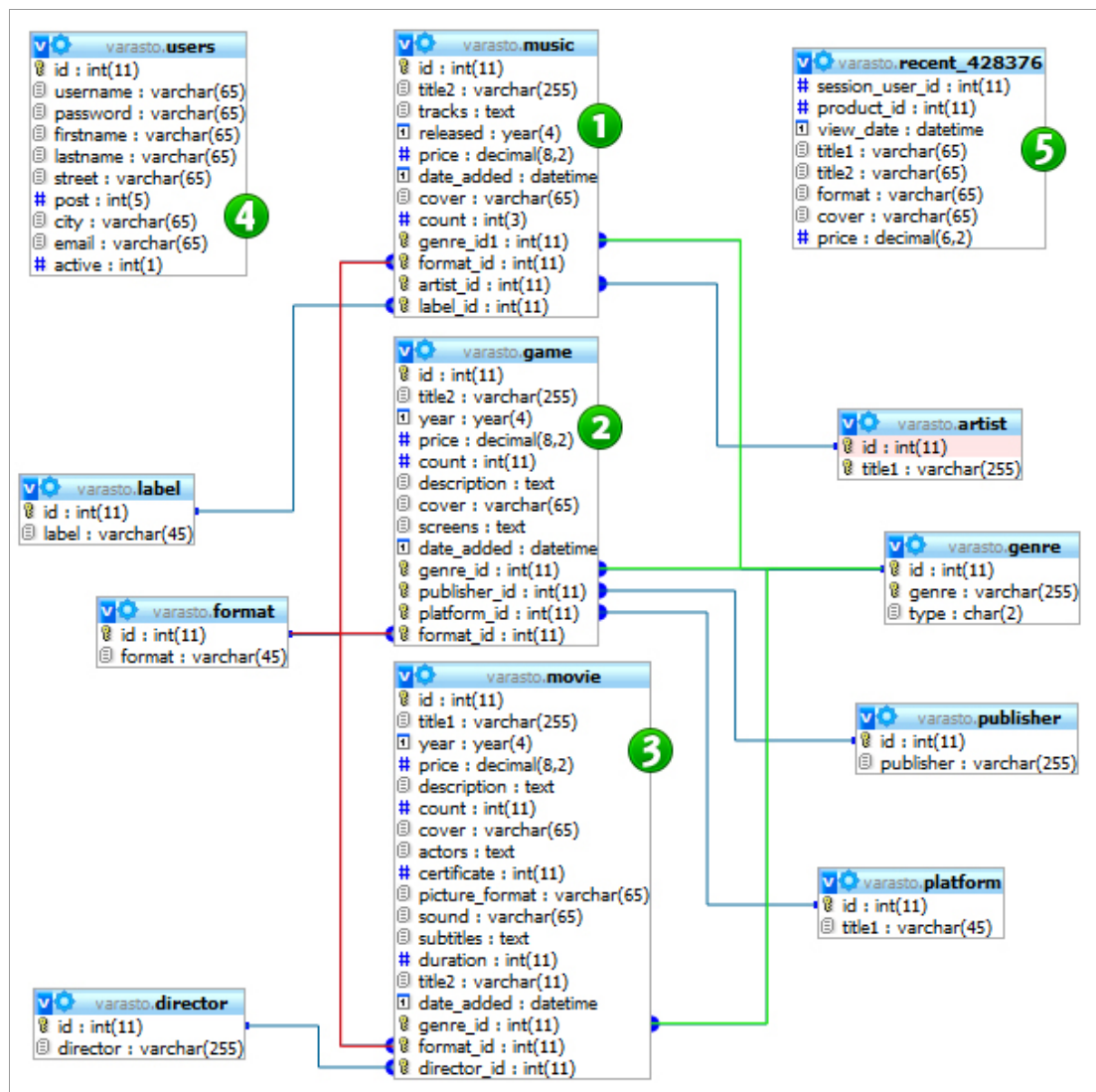
4.1.1 Relaatiotietokanta

Relaatiotietokannassa tietokokonaisuudet esitetään tauluina (table). Taulussa pystyriivejä kutsutaan sarakkeiksi (column) ja vaakarivejä riveiksi (row). Sarakkeita voidaan vaihtoehtoisesti kutsua myös kentiksi (field) ja rivejä tietueiksi (record). Taulun sisällä sarakkeilla on yksilölliset toisistaan poikkeavat nimet. Sarakkeen tiedot kuuluvat samaan arvojoukkoon (domain) ja niillä on sama tietotyyppi, kuten numeerinen tai merkkimuotoinen ja niille on varattu tietty pituus. Jokaisen taulun tunnisteenä on perusavain (primary key, PK). Perusavainsarakkeessa ei saa yhdessä taulussa olla kahta tai useampaa samaa arvoa vaan arvojen on oltava yksilöiviä, eli uniikkeja. Perusavain voi kuitenkin muodostua myös useammasta kuin yhdestä sarakkeesta. Taulussa ei myöskään saa olla tuplarivejä eikä toistuvia, alisteisia ryhmiä ja perusavainta käytetään

estämään tuplarivien muodostumista. Relaatiomallissa tietoja käsitellään joukko-opillisesti. Taulut muodostuvat joukosta rivejä joihin voi muodostaa voimakkaita joukkuoperaatioita, kuten ”hae kaikki vuonna 2007 julkaistut levyt (valinta), niistä esittäjä, nimike ja hinta (projektio)” (Hovi, Huotari, Lahdenmäki, 2005, 7-10).

4.2 Tietokannan rakenne

Aluksi suunnittelin paperilla mitä tauluja ja kenttiä verkkokaupan onnistuneessa toteutuksessa tulisin tarvitsemaan ja näiden suunnitelmien pohjalta sitten rakensin phpMyAdminilla seuraavanlaisen tietokannan:



Kuva 3. Tietokannan rakenne.

Tietokannassa on kolme päätaulua, joihin relaatioiden kautta kytkeytyy seitsemän pienempää taulua, joista id-numeroiden perusteella haetaan oikeat tiedot. Koska verkkokaupassa oli tarkoituksena myydä musiikkia, elokuvia ja pelejä, tarvittiin jokaiselle tuoteryhmälle omat taulunsa – music (1), game (2) ja movie (3). Esimerkiksi music -tauluun (1) tarvittiin tuotteen yksilöimiseksi kentät yksilölliselle, taulun perusavaimena toimivalle tunnistenumeralle (id), levyn nimelle (title2), kappaleiden nimille (tracks), julkaisuvuodelle (released), hinnalle (price), tietokantaan lisäysajankohdalle (date_added), kansikuvalla (cover), sekä varastosaldolle (count), joiden lisäksi erillisistä tauluista (format, label, artist ja genre) haettavat tiedot formaatista, levy-yhtiöstä, esittäjästä, sekä tyyllilajista. Päädyin erillisiin tauluihin suunnittelussa, koska esimerkiksi yhdellä artistilla voi olla monta erinimistä levyä, mutta yhdellä levyllä harvemmin on montaa eri esittäjää.

Movie ja game-taulut on tehty samalla periaatteella käyttäen elokuva- ja pelitallenteiden yksilöintiin välttämättömiä tietoja, kuten peleille julkaisijaa (publisher), alustaa (platform) ja kuvakaappauksia (screens), sekä elokuville ohjaajaa (director), näyttelijöitä (actors), kuvaformaattia (picture_format), ääniformaattia (sound), ikärajaa (certificate) ja tekstityksiä (subtitles). Molemmissa tauluissa on lisäksi kentät tuotekuvauksille (description) ja kansikuville (cover). Taulujen id-numeroille haluttiin vielä tehdä jako joka erottelee eri kategorian tuotteet toisistaan. Musiikki -kategorian tuotteiden id-numerot ovat väliltä 0-1000, elokuvien 1001-2000 ja pelien 2001-3000. Verkkokaupan kasvaessa id-alueet saattaisivat tietysti käydä liian pieniksi jos jossain kategoriassa olisi myynnissä yli 1000 tuotetta ja tämä pitäisi tietysti ottaa huomioon suunnittelussa, mutta tässä opinnäytetyön tapauksessa nämä rajat riittänevät.

Lisäksi tietokantaan tarvittiin vielä kaksi itsenäistä taulua, joilla ei ole relaatioita. Users-tauluun (4) tallennetaan rekisteröityjen käyttäjien tiedot, joita käytetään kassasivulla yhteystietojen noutamiseen. Kuvan recent_428376-tauluun (5) tallentuu käyttäjän viimeksi katsomat tuotteet. Taulun nimen numero-osa on satunnaisgeneraattorin luoma kuusinumeroinen tunniste käyttäjälle. Näistä lisää myöhemmin.

View Name	Columns
varasto.all_products	# id : int(11) @ title1 : varchar(255) @ title2 : varchar(255) # price : decimal(8,2)
varasto.all_music	# id : int(11) @ title1 : varchar(255) @ title2 : varchar(255) @ released : year(4) # price : decimal(8,2) @ tracks : text @ cover : varchar(65) # count : int(3) @ label : varchar(45) @ format : varchar(45) @ genre : varchar(255) @ date_added : datetime
varasto.all_game	# id : int(11) @ title2 : varchar(255) @ year : year(4) # price : decimal(8,2) @ description : text @ cover : varchar(65) # count : int(11) @ date_added : datetime @ screens : text @ format : varchar(45) @ genre : varchar(255) @ publisher : varchar(255) @ title1 : varchar(45)
varasto.all_movie2	# id : int(11) @ title1 : varchar(255) @ title2 : varchar(11) @ year : year(4) # price : decimal(8,2) @ description : text @ cover : varchar(65) # count : int(11) @ actors : text # certificate : int(11) @ picture_format : varchar(65) @ sound : varchar(65) @ subtitles : text # duration : int(11) @ date_added : datetime @ genre : varchar(255) @ director : varchar(255) @ format : varchar(45)

Kuva 4. Tietokannan koontitaulut.

Aikaisempien taulujen lisäksi tein vielä neljä erilaista koontitaulua, eli näkymää, joista kaikki sivuston tietokantaominaisuuksia hyödyntävät palvelut omat hakunsa tekevät. Näkymät ovat ikäänkuin virtuaalisia tauluja, ”tallennettuja hakuja”, joista voi niiden muodostamisen jälkeen tehdä hakuja aivan kuten tavallisista tauluista. Seuraavassa esimerkissä on tarvittava SQL-kysely all_music-näkymän muodostamiseen:

```
CREATE VIEW all_music AS
SELECT music.id, artist.title1, music.title2, music.released, music.price, music.tracks, music.cover,
music.count, label.label, format.format, genre.genre, music.date_added
FROM music, label, format, artist, genre
WHERE music.genre_id1 = genre.id
AND music.format_id = format.id
AND music.artist_id = artist.id
AND music.label_id = label.id
ORDER BY artist.title1
```

Esimerkki 9. Näkymän all_music luonti SQL:llä.

Esimerkin 9 ensimmäisellä rivillä luodaan näkymä ”all_music” (CREATE VIEW all_music AS) normaalista SQL-kyselystä, jossa valitaan music-taulusta kentät id, title2, released, price, tracks, cover, count ja date_added, sekä genre, format, artist ja label -tauluista kentät genre, format, artist ja label (SELECT music.id, artist.title1, music.title2, music.released, music.price, music.tracks, music.cover, music.count, label.label, format.format, genre.genre, music.date_added FROM music, label, format,

artist, genre), jotka vastaavat music-aulun arvoja genre.id, format.id, artist.id, sekä label.id (WHERE music.genre_id1 = genre.id AND music.format_id = format.id AND music.artist_id = artist.id AND music.label_id = label.id). Viimeisellä rivillä tulokset järjestetään esittäjien mukaan (ORDER BY artist.title1). Näin saadaan aikaiseksi ikään kuin yksi taulu (Kuva 5) joka sisältää kaikki tuotteen tiedot yhdessä paikassa ja josta on helppo suorittaa kaikki musiikkituotteisiin liittyvät haut.

id	title1	title2	released	price	tracks	cover	count	label	format	genre	date_added
12	Belphegor	Pestapokalypse IV	2006	12.95	Belphegor - Hell's Ambassador Seyn Todt In Sch...	images/covers/belphegor-pest.jpg	5	Nuclear Blast	CD	Death Metal	2011-02-22 00:00:00
15	Belphegor	Bondage Goat Zombie	2008	12.95	Bondage Goat Zombie Stigma Diabolicum Armag...	images/covers/belphegor-bondage.jpg	8	Nuclear Blast	CD	Death Metal	2011-02-22 00:00:00
13	Belphegor	Goatreich - Fleshcult	2005	12.95	The Cruzifixus - Anus Dei Bleeding Salvation ...	images/covers/belphegor-goat.jpg	4	Napalm	CD	Death Metal	2011-02-22 00:00:00
14	Belphegor	Lucifer Incestus	2003	8.95	Inflamate Christianos (Intro) The Goatchrist ...	images/covers/belphegor-lucifer.jpg	6	Napalm	CD	Death Metal	2011-02-22 00:00:00
5	Dark Tranquillity	The Gallery	1995	8.95	Punish My Heaven Silence, And The Firmament Wi...	images/covers/darktranquillity-gallery.jpg	6	Osmose Productions	CD	Melodic Death Metal	2011-02-22 00:00:00
6	Dark Tranquillity	The Mind's I	1997	8.95	Dreamlore Degenerate Zodiackyl Light Hedon...	images/covers/darktranquillity-minds.jpg	4	Osmose Productions	CD	Melodic Death Metal	2011-02-22 00:00:00

Kuva 5. Osa näkymän all_music sisällöstä.

5 RAKENNE JA TOIMINNALLISUUS

5.1 Sivuston rakenne

Viihdevarasto

Tervetuloa vieras! Kirjautu sisään | Rekisteröidy

Etusivu **Musiikki** Elokuvat Pelit

Kaikki tuotteet Hae Viihdevarastosta

Avajaistarjous!
Avajaisten kunniaksi kaikki tilaukset ilman postikuluja!
Tarjous voimassa rajoitetun ajan.

Ostoskori

- Origins of Plague Hateform
kpl à 15.95 €
- Stabbing the Drama Soilwork
1 kpl à 11.95 €
- The Panic Broadcast Soilwork
1 kpl à 18.95 €

Yhteensä: 30.90 €

Kassalle Tyhjennä

Musiikki näyttää lisää >

Machineae Supremacy Overworld CD 12.95 €	Dark Tranquillity Projector CD 8.95 €	Moonsorrow V. Hävityt CD 10.95 €	Dark Tranquillity Live Damage DVD 17.95 €

Elokuvat näyttää lisää >

The Pacific (6 levyä) Sota DVD 24.95 €	Narnian tarinat: Kaspianin matka maailman ääriin Fantasia Blu-Ray 21.90 €	Harry Potter ja kuoleman varjeluksen osa 1 Fantasia DVD 15.95 €	Inland Empire Draama DVD 8.90 €

Pelit näyttää lisää >

Gran Turismo 5 PS3 BluPlay 49.90 €	Killzone 3 PS3 BluPlay 59.90 €	Fable 3 XBOX 360 DVD 29.90 €	Dead Space 2 XBOX 360 DVD 59.90 €

Viimeksi katsotut tuotteet:

- Machinae Supremacy Redeemer 7.95 €
- Hevisaarus Jurahevin kuninkaast 9.95 €
- Soilwork Sworn to a Great Divide 12.95 €
- Soilwork The Panic Broadcast 18.95 €
- Soilwork Figure Number Five 8.95 €

Avajaisten kunniaksi kaikki tilaukset ilman postituskuluja!

© Oskari Väistö 2011

Kuva 6. Sivuston etusivu.

Kuvassa 6 näkyy verkkokaupan etusivu, jossa yläpalkista pääsee rekisteröitymään sivuston käyttäjäksi, sekä kirjautumaan sisään rekisteröidyttyään. Sivuston logon alla varsinaisessa valikkopalkissa on vasemmalla päävalikko josta aukeaa alavalikkoja pitämällä hiiren osoitinta pääsivujen valitsimien päällä (Kuvassa hiiren osoitin on Musiikki -pääsivun päällä ja siihen kuuluva alavalikko on auki). Valikkopalkin oikeassa reunassa on hakupalkki, jonka avulla käyttäjä voi suorittaa hakuja tietokannasta valitsemastaan kategoriasta ja hakea esimerkiksi vaikka kaikki elokuvat joiden nimessä on sana ”the”.

Etusivun oikeassa reunassa ovat allekkain ostoskori, sekä käyttäjän viimeksi katsomat tuotteet. Vasen reuna on varattuna varsinaiselle sisällölle ja esimerkikuvassa siinä näkyy joka osastolta neljä satunnaista tuotetta, sekä mainosbanneri.

Sivun oikea puoli pysyy sivuja vaihdettaessa samanlaisena, mutta oikealle puolelle ladataan aina tilanteen vaatimaa sisältöä PHP:llä:

```

1  if(isset($_GET['page'])) {
2
3      $pages=array("product", "products", "cart", "search", "registered", "login", "checkout",
4 "game", "movie", "logout", "main", "order", "music_main", "movies_main", "games_main",
5 "recovery", "mail");
6
7      if(in_array($_GET['page'], $pages)) {
8
9          $_page=$_GET['page'];
10
11      }else{
12
13          $_page="main";
14      }
15
16 }else{
17
18     $_page="main";
19
20 }
```

Koodi 1. index.php:lle sallittujen sisältösivujen määrittely.

Sivun index.php lohkon contentMain ladattaviksi sallittavat PHP-sivut (ilman .php-tarkennetta) on määritelty muuttujassa \$pages (rivillä 3) taulukkoon (array). Taulukko sisältää kaikki sivut, jotka index.php hyväksyy ladattavaksi niin pyydettyessä. Rivillä 1

tutkitaan onko selaimen osoiterivillä asetettu page-parametria (`if(isset($_GET['page']))`). Jos parametri on asetettu, tutkitaan sisätykö pyydetty sivu rivillä 3 määriteltyyn taulukkoon (`if(in_array($_GET['page'],$pages)`), rivillä 7). Jos parametri sisältyy taulukkoon, asetetaan muuttujan `$_page` arvoksi osoiterivin parametrin ”page” arvo komennolla `$_GET['page']` (rivi 9). Vastaavasti jos parametria ei löydy taulukosta, pakotetaan `index.php` lataamaan `contentMain`-lohkoon etusivun sisältö asettamalla muuttujan `$_page` arvoksi ”main” (rivi 13).

Muuttuja saa saman arvon ”main” myös siinä tapauksessa, ettei osoiterivillä ole määritelty page-parametria ollenkaan (rivi 18).

Parametri otetaan web-selaimen osoiteriviltä `$_GET['page']` -komennolla seuraavasti: jos osoiterivillä on osoite `../index.php?page=music_main`, ottaa komento `$_GET['page']` tässä tapauksessa talteen arvon ”music_main” eli musiikin pääsivun.

```

1 <div class="contentMain">
2
3     <?php require($_page.".php"); ?>
4
5 </div>
```

Koodi 2. Sisällön lataaminen `contentMain`-lohkoon `index.php`:ssa.

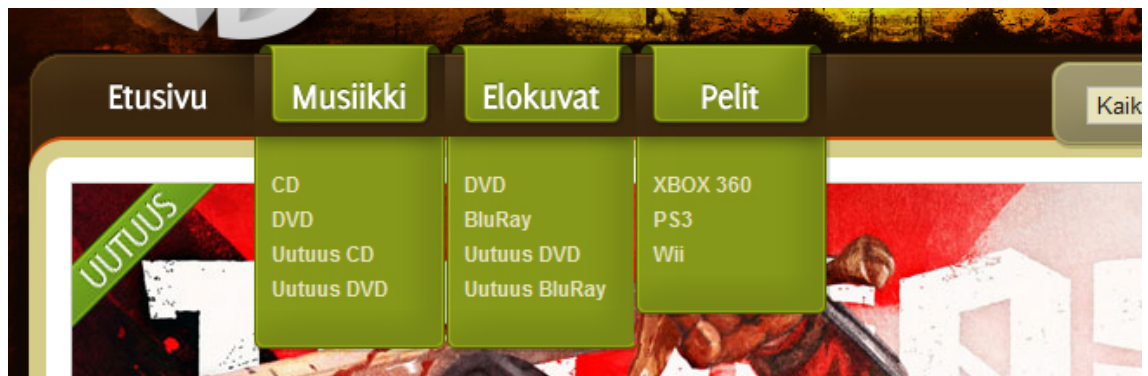
`Index.php`:n `contentMain`-lohkoon pyydetty sisällöt saadaan ladattua koodin 2 mukaisesti, jossa rivillä 3 `<?php require($_page.".php");?>` -koodilla ladataan muuttujaan `$_page` aiemmin koodissa 1 tallennettu arvo ja lisätään sen perään tarkenne (`.php`).

5.2 Toiminnallisuus

Sivustosta ja sen toiminnasta haluttiin selkeä ja helposti omaksuttava, jotta käyttäjän aika ei menisi epäloogisuuksien kanssa painimiseen, vaan hänen olisi helppo tehdä juuri sitä mitä hän on sivustolle tullutkin tekemään, eli keskittyä suorittamaan ostoksia.

5.2.1 Päävalikko

Päävalikko päädyttiin suunnitteluvaiheessa sijoittamaan vaakatasoon, koska silloin se vie sivuston muilta elementeiltä vähemmän tilaa leveyssuunnassa.



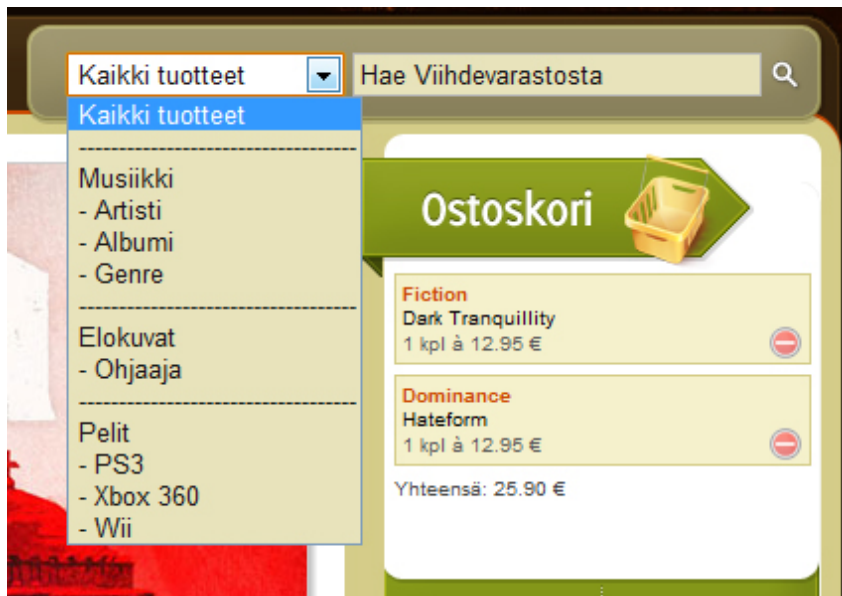
Kuva 7. Päävälikko (alavalikot auki).

Päävälikkona toimii tavallinen pudotus- eli ns. drop down -valikko, jossa hiiren osoittimen päävälikon päälle viemällä saa avatuksi alavalikon josta löytyy päävälikon arvoa tarkentavia vaihtoehtoja. Päävälikon tekstejä (Etusivu, Musiikki, Elokuvat ja Pelit) klikkaamalla pääsee kunkin tekstin osoittaman alueen pääsivulle, tai vaihtoehtoisesti sivuston etusivulle. Alavalikoista käyttäjä voi halutessaan saada esiin vaikka kaikki musiikki CD:t tai kaikki pelit XBOX 360 -konsolille.

Valikoiden näyttävyyden lisäämiseksi niihin lisättiin animointia jQueryllä, eli alavalikot eivät vain hiiren osoittimen päälle osuessa ilmesty tyhjästä tai vastaavasti pois päältä siirtyessä katoa näkyvistä, vaan ne liukuvat alas ja ylös sulavasti.

5.2.2 Haku

Hakupalkki (kuva 8) sijaitsee sivuston oikeassa ylänurkassa ja mahdollistaa hakujen suorittamisen MySQL-tietokantaan. Hakupalkin vasemmalla alavetovalikolla käyttäjä voi tarkentaa tekemäänsä hakuja koskemaan esimerkiksi vain tietyn artistin levyjä tai vain tietyn ohjaajan elokuvia.



Kuva 8. Hakupalkki.

Hakupalkki on toteutettu CSS:llä koristetulla HTML-kaavakkeella (form), joka välittää käyttäjän tekemät valinnat sivulle `search.php` ”source”-nimisen valintalistan ja tekstikentän ”term” arvoina.

Hakukentän lähtötilanteen arvo ”Hae Viihdevarastosta” saadaan käyttäjän valitessa sen hiirellä tyhjenemään JavaScriptin avulla (`onClick = this.value=""`”). Lause siis asettaa hiiren klikkauksella tekstikentän arvoksi tyhjän eli tyhjentää sen.

Kuvitellaan, että tehdään haku kirjoittamalla tekstikenttään ”soil” ja valitsemalla valintalistasta kohta ”Artisti”:

```

1  if (isset($_POST['term'])) {
2      $term = $_POST['term'];
3      $source = $_POST['source'];
4  }
5
6  switch ($source) {
7      case "artist":
8          $myquery = "SELECT * FROM all_music WHERE all_music.title1
9                      LIKE '%$term%'";
10         break;
11     }
12
13 $finalquery = $myquery.$pricefilter.$genrefilter.$formatfilter;
14
15 $sql = mysql_query($finalquery) or die(mysql_error());
16 $count = mysql_num_rows($sql);

```

```

17 switch ($count) {
18     case 0:
19         echo '<div class="result2">Hakusi <p id="orange">'. $term. '</p> ei
20             tuottanut tuloksia.</div>';
21         break;
22     case 1:
23         echo '<div class="result2">Hakusi <p id="orange">'. $term. '</p> tuotti
24 osuman '.$extension.';</div>';
25         break;
26     case ($count > 1):
27         echo '<div class="result2">Hakusi <p id="orange">'. $term. '</p> tuotti '.
28 osumaa '.$extension.';</div>';
29         break;
30 }
31
32 while ($row = mysql_fetch_array($sql)){
33     $found = true;
34     $cover = $row[cover];
35     $format = $row[format];
36     ?>
37     <div class="result">
38     <?php
39         switch ($row[id] {
40             case ($row[id] < 1000): //musiikkia ?>
41                 <a class="search" href="index.php?page=product&product_id=
42                     <?php echo $row[id] ?>">
43                     <div class="artist"><?php echo $row[title1]?></div>
44                     <div class="title"><?php echo $row[title2]?> |
45                     <div class="format"><?php echo $row[format]?></div>
46                     </div>
47                     <div class="price"><?php echo $row[price]?> €</div>
48                     <div class="image">
49 <?php
50                         switch ($format) {
51                             case "CD":
52                                 echo '';
53                                 break;
54                             case "DVD":
55                                 echo '';
56                                 break;
57                         }?>
58                     </div>
59                     </a>
60 <?php
61                         break;
62                     }
63     ?>
64     </div> <!-- end result -->
65 <?php
66 }
67 ?>
68

```

Koodi 3. Hakutietojen eteneminen search.php:ssä.

Edellä annetuilla hakuehdoilla tutkitaan ensiksi onko hakutekstiä lähetetty (rivi 1, if (isset(\$_POST['term']))) ja asetetaan sen jälkeen hakulomakkeella post-metodilla lähetetyt tekstikentän arvo ”term” ja valintalistan arvo ”source” muuttujiksi \$term ja \$source (rivit 2 ja 3).

Seuraavaksi tutkitaan muuttujan \$source arvoa PHP:n switch – case lauseparin avulla (rivit 6 - 11). Esimerkin tapauksessa \$source:n arvon ollessa ”artist” asetetaan muuttujan \$myquery arvoksi SQL-kysely ”SELECT * FROM all_music WHERE all_music.title1 LIKE '%term%'” eli selkokielellä: ”hae kaikki näkymän ”all_music” rivit joissa sarake ”title1” (artisti) sisältää hakusanan ”soil”. Rivillä 13 muuttujan \$myquery perään lisätään mahdolliset (myöhemmin search.php sivulta valittavat) tulosten suodatusehdot ja tallennetaan näin syntynyt uusi muuttuja nimellä \$finalquery.

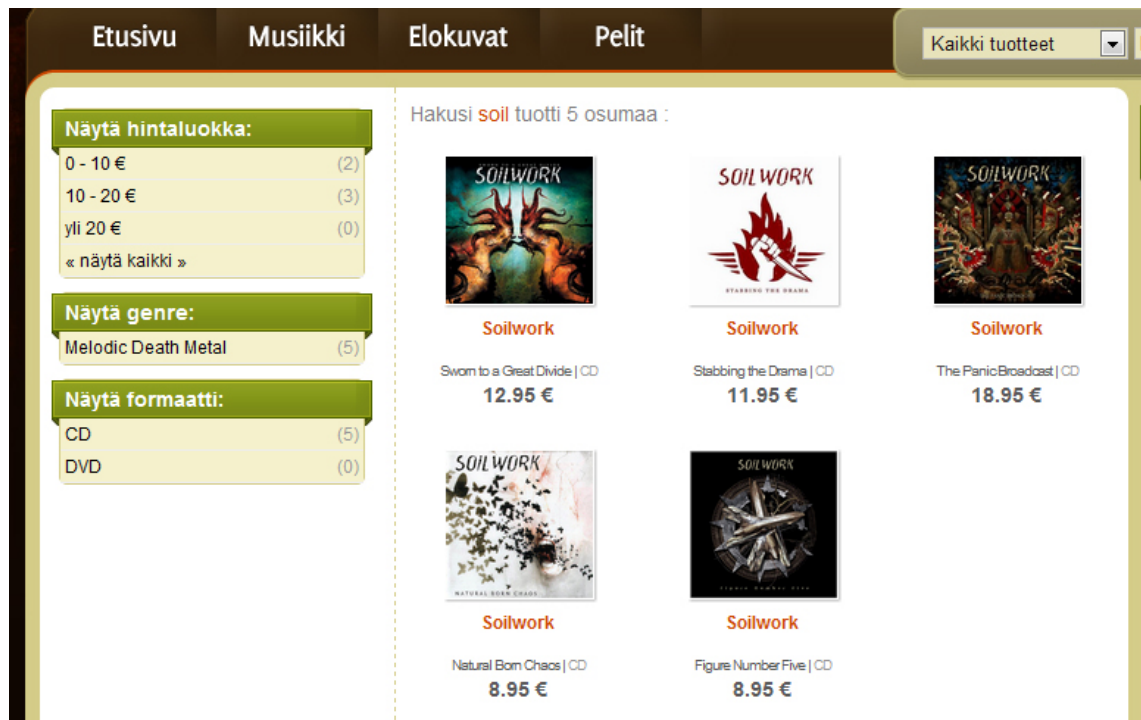
Seuraavaksi tällä uudella muuttujalla suoritetaan funktiolla mysql_query(\$finalquery) SQL-kysely tietokantaan rivillä 15 ja rivillä 16 talletetaan funktiolla mysql_num_rows(\$sql) saatujen tulosten lukumäärä muuttujaksi \$count. Tulosten lukumäärän mukaan tulostetaan näytölle joko löytyneiden tulosten lukumäärä, tai vaihtoehtoisesti se, ettei näillä hakuehdoilla löytynyt yhtään tulosta. Esimerkin tapauksessa tuloksia on enemmän kuin yksi, joten näytölle tulostuu switch – case lauseparin vaihtoehto \$count > 1:n (rivi 26) määrittelemä teksti ”Hakusi soil tuotti 5 osumaa :” (rivit 27 - 28).

Rivillä 32 määritellään \$row – muuttujan arvoksi mysql_fetch_array -funktiolla taulukko, joka sisältää hakuehtojen tuottamat tulokset riveinä. Samalla rivillä alkavalla while-lauseella tulostetaan hakutuloksia rivi kerrallaan niin kauan kuin taulukossa riittää rivejä. Riveillä 34 ja 35 tallennetaan \$row-muuttujaan tallennetun sarakkaiden cover ja format arvot omiksi muuttujikseen \$cover ja \$format. Rivin 39 switch – case lauseella erotellaan tuotteet vielä id-numeroiden perusteella musiikkiin, elokuviin tai peleihin. Esimerkkitapauksessa etsittiin vain tietyn artistin tuotteita, joten kaikki tulokset osuvat musiikkikategoriaan ja muut vaihtoehdot on koodista poistettu.

Riveillä 41 ja 42 muodostetaan yhdestä hakutuloksesta linkki (<a class="search" href="index.php?page=product&product_id=<?php echo \$row[id] ?>">), jota klikkaamalla pääsee tuotesivulle, jonne kyseisen tuotteen tiedot aukenevat linkissä olevan product_id -numeron perusteella. Linkin sisälle tulostetaan esimerkkitapauksessa artistin nimi (\$row[title1]), tuotteen nimi (\$row[title2]), formaatti (\$row[format]), hinta

(\$row[price]) ja kansikuva (\$row[cover]). Koska eri formaateissa (CD ja DVD) kansikuvat ovat erikokoiset, tarvitaan vielä yksi switch – case -lause, joka muuttujan \$format perusteella tulostaa kansikuvan kunkin formaatin vaatimin CSS-määrittelyin.

Seuraavassa esimerkkitapauksen hakuehtojen tuottama tulossivu:



Kuva 9. Hakutulokset ehdoilla Artisti, ”soil”.

Hakutuloksia voi suodattaa edelleen sivun vasemmassa reunassa olevilla suodattimilla esimerkiksi seuraavasti:

Painetaan hintaluokan 0 – 10€ -filteriä. Linkissä on koodi `index.php?page=search&filterbyprice=1`.

```

1 $filterbyprice = $_GET['filterbyprice'];
2 if (isset($filterbyprice)) {
3     switch ($filterbyprice){
4         case 0:
5             $pricefilter = " AND price > 0";
6             break;
7         case 1:
8             $pricefilter = " AND price BETWEEN 0 AND 10";
9             $extension = " hintaluokassa 0 - 10 euroa";
10            break;
11     }
12 }

```

```

13
14 $finalquery = $myquery.$pricefilter.$genrefilter.$formatfilter;
15
16 $sql = mysql_query($finalquery) or die(mysql_error());
17 $count = mysql_num_rows($sql);
18 switch ($count) {
19     case ($count > 1):
20         echo '<div class="result2">Hakusi <p id="orange">'. $term.' </p>
21             tuotti '. $count.' osumaa '.$extension.';</div>';
22         break;
23 }

```

Koodi 4. Hintasuotimen toiminta.

Rivillä 1 tallennetaan osoitepalkin osoitteesta parametrin filterbyprice arvo 1 muuttujaan \$filterbyprice. Jos parametri on asetettu, tutkitaan sen arvoa switch – case -lauseella (rivi 3). Esimerkkitapauksessa arvon ollessa 1, muuttuja \$pricefilter saa arvon ” AND price BETWEEN 0 AND 10” (rivi 8) ja muuttuja \$extension arvon ” hintaluokassa 0 – 10 euroa” (rivi 9). Lopuksi muutetaan äskeisessä esimerkissäkin ollutta \$myquery-muttujaa (SELECT * FROM all_music WHERE all_music.title1 LIKE '%soil%') lisäämällä sen perään suodinmuuttujista hintasuodin \$pricefilter (koska osoitteen mukana ei ole välitetty muita parametreja kuin filterbyprice ovat muut muuttujat tyhjiä ja näin ollen ainoastaan hintasuodin vaikuttaa lopputulokseen), jolloin lopullinen tuotteiden hakuun käytetty muuttuja \$finalquery (rivi 14) saa muodon \$myquery + \$pricefilter eli ”SELECT * FROM all_music WHERE all_music.title1 LIKE '%soil%' AND price BETWEEN 0 AND 10”. Selkokielisesti ”hae taulusta all_music kaikki rivit, joissa esittäjän nimi sisältää sanan 'soil' ja jonka hinta on väliltä 0 – 10 euroa”.

Muuttujaa \$extension käytetään tulostamaan käyttäjälle tekstin: ”Hakusi soil tuotti 2 osumaa hintaluokassa 0 - 10 euroa: ” riveillä 21 ja 22.

5.2.3 Ostoskori

Ostoskori päädyttiin toteuttamaan PHP:n istunnonhallintaominaisuuksilla. Käytännössä tämä tarkoittaa sitä, että ostoskori on voimassa istunnon ajan eli katoaa muistista käyttäjän lopettaessa session tai sammuttaessa selaimen.

Ostoskoriin tuotetta lisättäessä ilmestyy aina jokaiselle tuotteelle oma reunuksella rajattu laatikkonsa sisältäen tuotteen tiedot. Ensimmäisellä rivillä on tuotteen nimi,

toisella tuotteen tyypistä riippuen joko esittäjä, formaatti tai pelialusta ja kolmannella ostoskorissa olevien samojen tuotteiden lukumäärä, sekä tuotteen kappalehinta. Tuotelaatikoiden alle tulostetaan vielä ostoskorissa olevien tuotteiden yhteishinta. Lisäksi jokaisen tuotelaatikon oikeassa alakulmassa on pieni punainen nappi, jota klikkaamalla käyttäjä poistaa tuotteita korista joko pienentämällä lukumäärää yhdellä tai jos korissa on vain yksi kappale kyseistä tuotetta, poistamalla sen kokonaan korista.

Vihreästä alapalkista on mahdollisuus joko siirtyä tekemään tilausta painamalla ”kassalle”-nappia tai tyhjentää koko ostoskorinsa painamalla ”tyhjennä”-nappia. Napit ovat aktiivisina vain jos korissa on tuotteita.



Kuva 10. Ostoskori.

```

1 session_start();
2
3 if(isset($_GET['action']) && $_GET['action']=="add") {
4     $id=$_GET['id'];
5     if(isset($_SESSION['cart'][$id])) {           //tuote löytyy sessiosta
6         $_SESSION['cart'][$id]['quantity']++;
7     }else{                                       //ei löydy
8         $sql_s="SELECT * FROM $table WHERE id='$id'";
9         $query_s=mysql_query($sql_s);
10        if(mysql_num_rows($query_s) !=0) {       //product_id löytyy!
11            $row_s=mysql_fetch_array($query_s);
12            $_SESSION['cart'][$row_s['id']]=array(
13                "quantity" => 1,
14                "price" => $row_s['price']
15            );
16        }else{
17            $message="Virheellinen tuote-id!";
18        }
19    }

```

20 }

Koodi 5. Tuotteen lisääminen ostoskoriin.

Istunto eli sessio aloitetaan aina `session_start()` -funktiolla (rivi 1). `Session_start()` funktiota tulee kutsua aina jokaisen sivun alussa erikseen, jos tarvitsee sivun toiminnoissa sessiomuuttujia. Tuotteen ostoskoriin lisäämiskäskey välitetään `shoppingCart.php`:lle tuotesivujen ”osta” -nappien linkkien välityksellä esimerkiksi seuraavasti: ”`index.php?page=product&product_id=54&action=add&id=54`”, jolloin koodin x rivillä 3 tutkitaan onko parametria ”action” asetettu ja onko parametrin arvo ”add”. Esimerkkitapauksessa on, joten siirrytään riville 4 ja tallennetaan muuttujan `$id` arvoksi `get`-metodilla osoiterivin kohdasta ”`&id=54`” id:n arvo eli numero 54. Seuraavaksi tutkitaan (rivillä 5) löytyykö istunnon taulukosta jo riviä `$_SESSION['cart']['id']`. Jos rivi löytyy, kasvatetaan tuotteen lukumäärä taulukossa yhdellä (rivi 6).

Jos tuotetta ei vielä löydy sessiosta, tutkitaan löytyykö `$id`:tä vastaavaa tuotetta tietokannasta (rivit 8-9). `$table` -muuttuja on joko `all_music`, `all_game`, tai `all_movie2`, riippuen jo aikaisemmin tietokantojen yhteydessä kerrotuista id-rajoista (id:llä 54 se on `all_music`). Jos haun tuloksena saatujen rivien lukumäärä on erisuuri kuin nolla (rivi 10), eli tuote löytyy tietokannasta, luodaan taulukkoon `$_SESSION['cart']` uusi rivi. Indeksinä käytetään SQL-kyselyllä saatua `$_row['id']`:n arvoa. Rivin sarakkeiden ”quantity” (lukumäärä) ja ”price” (hintaa) arvoiksi asetetaan 1 ja tietokantakyselyn tuloksena saatu tuotteen hinta `$row_s['price']`.

```

1  if(isset($_GET['action']) && $_GET['action']=="remove") {
2
3      $id=intval($_GET['id']);
4      $qty=$_SESSION['cart'][$id]['quantity'];
5
6      switch($qty){
7          case($qty > 1):
8              $_SESSION['cart'][$id]['quantity']--;
9              break;
10         case 1:
11             unset($_SESSION['cart'][$id]);
12             break;
13     }
14 }
```

Koodi 6. Tuotteiden vähentäminen ostoskorista.

Tuotteiden vähentäminen ostoskorista on toteutettu samaan tapaan kuin äskeinen tuotteen lisääminen ostoskoriin. Tällä kertaa tutkitaan onko parametrin ”action” arvo ”remove” (rivillä 1). Jos näin on ja korissa on tuotetta enemmän kuin yksi kappale, pienennetään lukumäärää yhdellä (rivi 8). Tuotteiden lukumäärän ollessa yksi poistetaan tuote kokonaan (rivi 11) ostoskorista.

5.2.4 Viimeksi katsotut

Lista käyttäjän viimeksi katsomista tuotteista päädyttiin toteuttamaan memory-varastointimoottorilla tietokantaan tallennettavilla väliaikaisilla tauluilla. Memory-tyyppiset taulut tallennetaan nimensä mukaisesti muistiin ja niiden sisältämät rivit pyyhkiytyvät tietokannasta aina kun tietokantapalvelin käynnistetään uudelleen.



Kuva 11. Viimeksi katsotut tuotteet

```

1 <?php
2 session_start();
3 $pid = $_GET[product_id];
4 if(!$SESSION[user_id]){
5     $SESSION[user_id] = rand(1, 100000); //luodaan session_user_id
6 }
7
8 if (isset($pid) and $pid != ""){
9     switch ($pid) {
10         case ($pid < 1000): //musaa
11             $table = "all_music";
12             break;
13         case ($pid >= 1000 and $pid < 2000): //leffoja
14             $table = "all_movie2";
15             break;
16         case ($pid >= 2000): //pelejä
17             $table = "all_game";
18             break;
19     }
20     $productquery = "SELECT * FROM $table WHERE id = $pid";
21     $products = mysql_query($productquery);
22     $row = mysql_fetch_array($products);
23     $title1 = $row[title1];
24     $title2 = $row[title2];
25     $cover = $row[cover];
26     $price = $row[price];
27     $format = $row[format];
28
29     $sql = ("CREATE TABLE IF NOT EXISTS recent_$SESSION[user_id] (
30         session_user_id INT(11) NOT NULL,
31         product_id INT(11) NOT NULL,
32         view_date DATETIME NOT NULL,
33         title1 VARCHAR(65) NOT NULL,
34         title2 VARCHAR(65) NOT NULL,
35         format VARCHAR(65) NOT NULL,
36         cover VARCHAR(65) NOT NULL,
37         price DECIMAL(6,2) NOT NULL)
38         ENGINE = MEMORY;");
39     mysql_query($sql);
40
41     $sql2 = mysql_query("SELECT * FROM recent_$SESSION[user_id] ORDER BY view_date
42     DESC LIMIT 1"); //haetaan uusin tuote
43     $row2 = mysql_fetch_array($sql2);
44     $pid2 = $row2[product_id];
45
46     if ($pid2 != $pid) { //lisätään vain jos uusin ei ole sama tuote
47
48         mysql_query("INSERT INTO recent_$SESSION[user_id] (
49             session_user_id,
50             product_id,
51             view_date,
52             title1,
53             title2,
54             format,
55             cover,
56             price)
57             VALUES (
58             '$SESSION[user_id]',
59             '$pid',
60             now(),
61             '$title1',
62             '$title2',

```

```

63         '$format',
64         '$cover',
65         '$price');"
66     }
67 }
68 ?>

```

Koodi 7. Viimeksi katsotut tuotteet.

Rivillä 4 tutkitaan onko istuntomuuttujalle \$_SESSION[user_id] asetettu arvoa. Jos ei ole, arvotaan PHP:n RAND() -funktiolla muuttujalle arvo väliltä 1 – 1 000 000 (rivi 5). Loput koodista ajetaan vain jos muuttuja \$pid (product id) on asetettu eikä sen arvo ole tyhjä. Tällä pyritään estämään mahdolliset virhetilanteet. Seuraavaksi asetetaan muuttujalle \$table arvo lisättävän tuotteen tuote_id:n mukaan (rivit 9 – 19). Muuttujaa käytetään hyväksi rivillä 20 kohdistamaan SQL-haku oikeaan tauluun. Haun tulokset tallennetaan omiin muuttujiinsa riveillä 23 – 27. Riveillä 29 – 39 luodaan taulu recent_xxxxxxx, jonka nimen loppuosan muodostaa alussa arvoitun \$_SESSION[user_id]:n arvo ("CREATE TABLE IF NOT EXISTS recent_\$_SESSION[user_id]"). Taulu luodaan vain, ellei saman nimistä taulua ole jo olemassa. Taulu määrätään käyttämään memory -varastointimoottoria (ENGINE = MEMORY, rivi 38). Riveillä 41-44 haetaan tästä uudesta väliaikaisesta taulusta siihen viimeiseksi lisätyn tuotteen id, jota sitten verrataan seuraavaksi lisättävän tuotteen tuote-id:hen rivillä 46 ja uusi tuote lisätään listalle (rivit 48 - 65) vain jos se on eri tuote kuin viimeksi lisätty. Tällä haluttiin estää saman tuotteen ilmestymisen listalle useammin kuin kerran peräkkäin.

Seuraavaksi tuotteet tulostetaan väliaikaisesta taulusta näytölle:

```

69 $checkquery ="SELECT * FROM information_schema.tables
70             WHERE table_schema = 'varasto'
71             AND table_name = 'recent_$_SESSION[user_id]";
72 $checkresult = mysql_query($checkquery);           //tutkitaan onko taulu olemassa
73 $num = mysql_num_rows($checkresult);
74
75 if (isset($_SESSION[user_id]) and $num != 0) {    //tehdään haku vain jos taulu on olemassa
76     $found = false;
77     $query = "SELECT * FROM recent_$_SESSION[user_id] WHERE session_user_id =
78             '$_SESSION[user_id]' ORDER BY view_date DESC LIMIT 6";

```

Koodi 8. Viimeksi katsottujen tuotteiden tulostus.

Aluksi tehdään kysely (rivit 69 – 72) tietokantaserverin kaikista tauluista tiedot sisältävään information_schemaan (SELECT * FROM information_schema.tables) ja tarkemmin tietokantaan "varasto" (WHERE table.schema = 'varasto') ja tauluun

recent_XXXXXXX (AND table_name = 'recent_\${_SESSION[user_id]}'). Rivillä 73 tallennetaan saatujen osumien lukumäärä muuttujaan \$num. Seuraavaksi tarkastetaan ehtolauseella, että \$_SESSION[user_id] on olemassa (if(isset(\$_SESSION[user_id]))) ja äsken saatujen hakutulosten lukumäärä ei ole nolla (and \$num != 0) eli taulu on olemassa. Jos taulu löytyy niin tehdään sinne haku (rivit 77 ja 78) käyttäjätunnuksella (SELECT * FROM recent_\${_SESSION[user_id]} WHERE session_user_id = '\${_SESSION[user_id]}') ja järjestetään rivit niiden iän mukaan (ORDER BY view_date) laskevaan järjestykseen (DESC). Hakutulokset rajoitetaan kuuteen (LIMIT 6), jolloin saaduista hakutuloksista saadaan tulostettua sivulle 5 uusinta riviä.

5.2.5 Pääsivut

UUTUUS

Tutrisas - STAND UP AND FIGHT! Sota jatkuu, hyppää mukaan! Uusi albumi ulkona nyt!

1 2 3 4

Viimeksi lisätyt [näytä lisää >](#)

Scar Symmetry Fitch Black Progress CD 8.95 €	Scar Symmetry SymmetricIn Design CD 8.95 €	Scar Symmetry Holographic Universe CD 10.95 €	Scar Symmetry Dark Matter Dimensions CD 12.95 €

CD [näytä lisää >](#)

Belphegor Pestapokalypse IV CD 12.95 €	Dark Tranquillity Projector CD 8.95 €	In Flames Soundtrack to Your Escape CD 8.95 €	Scar Symmetry SymmetricIn Design CD 8.95 €

DVD [näytä lisää >](#)

Opeth The Roundhouse Tapes DVD 12.95 €	Riverside Reality Dream DVD 23.95 €	Opeth Lamentations DVD 8.95 €	Dark Tranquillity Live Damage DVD 17.95 €

Kuva 12. Musiikkiosion pääsivu.

Jokaisella tuoteryhmällä (musiikki, elokuvat ja pelit) on lähes identtiset pääsivut joista esimerkkinä ylempänä musiikkiosion pääsivu (kuva 12). Pääsivuilla ylempänä on mainosbanneri, jolla tuotteita voi nostaa ns. näytetikkunaan. Banneri on toteutettu jQueryllä tehdyllä Slidebox Banner Rotatorilla, eikä siinä ole juuri muuta minun tekemääni kuin ulkoasun sovittaminen sivuston ulkoasuun ja itse mainoskuvat teksteineen. Halusin lisätä sen kuitenkin pääsivuille lähinnä visuaalisista syistä.

Bannerin alla seuraavana on kohta ”viimeksi lisätyt”, jossa näkyy 4 viimeksi tietokantaan lisättyä musiikkiosion tuotetta. ”Viimeksi lisätyjen” alla on neljä satunnaista esimerkkiä sekä CD-, että DVD-valikoimasta. Kaikkien kohtien otsakepalkkien oikeassa reunassa on lisäksi ”näytä lisää” -painikkeet, joita painamalla pääsee katsomaan esimerkiksi kaikkia tietokannan CD-levyjä.

5.2.6 Tuotesivut

Menupalkin linkkejä klikkaamalla pääsee kunkin aihealueen tuotesivuille, joilla olevat tiedot käyttäjä voi halutessaan järjestää haluamaansa järjestykseen Javascriptin *onchange*-ominaisuutta hyväkseen käyttävillä pudotusvalikoilla. Käytännössä tämä toimii siten, että käyttäjän muuttaessa pudotusvalikon arvoa, ladataan sivu uudelleen ja lähetetään PHP:llä osoitepalin kautta tiedot tehdyistä muutoksista. Tiedot pystyy järjestämään vaikka esittäjän, nimikkeen tai hinnan mukaiseen järjestykseen.

Musiikki CD

Järjestä: Nimikkeitä per sivu:

1 | 2 | 3 | 4 »

			
Machinae Supremacy	Riverside	Belphegor	Dark Tranquillity
A View From The End Of The World CD 16.95 €	Anno Domini High Definition CD 15.95 €	Bondage Goat Zombie CD 12.95 €	Charader CD 12.95 €
			
In Flames	Dark Tranquillity	Hateform	Dark Tranquillity
Colony CD 8.95 €	Damage Done CD 11.95 €	Dominance CD 12.95 €	Fidion CD 12.95 €
			
Soilwork	Belphegor	Dark Tranquillity	Hevisaurus
Figure Number Five CD 8.95 €	Goatrich - Fleshalt CD 12.95 €	Haven CD 8.95 €	Himuläisöjen yö CD 15.90 €

1 | 2 | 3 | 4 »

Kuva 13. Tuotesivu, musiikki – cd.

Tuotesivuilla näytölle tulostetaan nimi- ja hintatiedot kansikuvineen kulloinkin tarkastelun alla olevan aihealueen tuotteista neljän tuotteen riveinä. Käyttäjällä on mahdollisuus valita oikean puoleisesta pudotusvalikosta kerralla näytettävien tuotteiden lukumäärä 12 inkrementteinä väliltä 12 – 60 (eli näytettäviä tuoterivejä 3 – 15). Aihealueen käyttäjä näkee vihreän yläpalkin päädyissä olevista kuvilla toteutetuista teksteistä.

Tuotetietojen ylä ja alapuolelle tulostetaan vielä tarvittaessa (jos tuotteita on enemmän kuin kulloinkin valittuna oleva kerralla sivulla näytettävien nimikkeiden lukumäärä)

pystyviivoin toisistaan eroteltu sivunumerolistaus, josta käyttäjä pääsee numeroa klikkaamalla siirtymään kyseiselle sivulle tai vaihtoehtoisesti numerolistan päässä olevia nuolia klikkaamalla joko edelliselle tai seuraavalle numerolistan sivulle. Kuvassa 13 ollaan ensimmäisellä sivulla joten nuolta edelliselle sivulle ei näytetä. Nykyinen sivu kerrotaan käyttäjälle oranssilla ja lihavoidulla numerolla.

The screenshot shows a music store interface for the album 'Second Life Syndrome' by Riverside. The main product is displayed with its cover art, price (11.95 €), and a 'Osta' (Buy) button. Below the price, it lists the release year (2005), format (CD), and quantity (2 kpl). To the right, a 'Kappaleet' (Tracks) list shows 9 songs. Further right, a 'Muista myös nämä:' (Also remember these:) section features four recommended albums with their covers, titles, and prices. At the bottom, a 'Lisää Riverside -tuotteita' (More Riverside products) section displays four related items: 'Anno Domini High Definition', 'Rapid Eye Movement', 'Second Life Syndrome', and 'Reality Dream', each with its cover, title, format, and price.

RIVERSIDE
Second Life Syndrome
 [InsideOut]

Kappaleet

01. After
02. Volte-Face
03. Conceiving You
04. Second Life Syndrome
05. Artificial Smile
06. I Turned You Down
07. Reality Dream III
08. Dance With The Shadow
09. Before

Muista myös nämä:

- Von Herten Brothers - Love Remains The Same 8.95 € **Osta**
- Riverside - Rapid Eye Movement 12.95 € **Osta**
- Von Herten Brothers - Stars Aligned 17.95 € **Osta**
- Porcupine Tree - The Incident 15.95 € **Osta**

Lisää Riverside -tuotteita

- Anno Domini High Definition | CD 15.95 €
- Rapid Eye Movement | CD 12.95 €
- Second Life Syndrome | CD 11.95 €
- Reality Dream | DVD 23.95 €

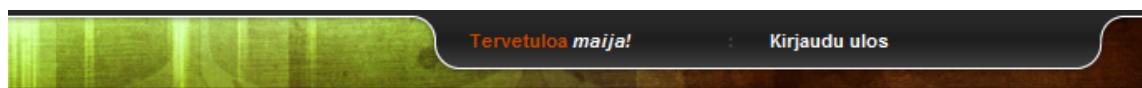
Kuva 14. Yksittäisen cd:n tuotesivu.

5.2.7 Rekisteröityminen ja kirjautuminen

Sivuston yläpalkkia klikkaamalla palkki liukuu alas ja antaa asiakkaalle mahdollisuuden rekisteröityä sivuston käyttäjäksi. Tällöin hänen ei sivustolla asioidessaan ja tilausta jättäessään tarvitse täyttää tietojaan tilauslomakkeeseen aina uudelleen vaan tarvittavat tiedot löytyvät tietokannasta.

Kuva 15. Rekisteröinti- ja kirjautumisenäkymä.

Samasta paikasta pääsee jo aikaisemmin rekisteröitynyt käyttäjä kirjautumaan sivustolle omilla tunnuksillaan.



Kuva 16. Yläpalkki sisäänkirjautumisen jälkeen.

Kun käyttäjä on kirjautuneena sivustolle, muuttuu yläpalkin ”Tervetuloa vieras!” -tekstin sanan ”vieras” tilalle käyttäjän käyttäjätunnus ja sisäänkirjautumislinkki muuttuu uloskirjautumislinkiksi.

5.2.8 Salasan palautus

Jos käyttäjä on unohtanut tunnuksensa, saa ne palautettua sähköpostiinsa ”Tunnukset hukassa?” -linkkiä klikkaamalla ja antamalla käyttäjätunnuksensa ja siihen kuuluvan sähköpostiosoitteen. Koska tietokannassa ei säilytetä mitään kovin salaista tietoa, kuten luottokortin numeroa, tämän tietoturvatason katsottiin riittävän tässä työssä.

5.2.9 Kassatoiminnot

Kun asiakas on lisännyt tuotteita ostoskoriinsa, muuttuu ostoskorisovelluksen vasemman alareunan ”Kassalle” -linkki aktiiviseksi ja sitä klikkaamalla pääsee tutkimaan ostoskorin sisältöä tarkemmin.

Ostoskorin sisältö:

Tuote	Kpl	à hinta	Hinta
Draama - Inland Empire	<input type="text" value="1"/> x	8.90 € =	8.90 €
Machinae Supremacy - Overworld	<input type="text" value="1"/> x	12.95 € =	12.95 €
Machinae Supremacy - A View From The End of The World	<input type="text" value="1"/> x	16.95 € =	16.95 €
XBOX 360 - Fable 3	<input type="text" value="2"/> x	29.90 € =	59.80 €
			Yhteensä > 98.60 €

Poista tuote asettamalla määrä nollassi.

Kuva 17. Ostoskorin sisältö.

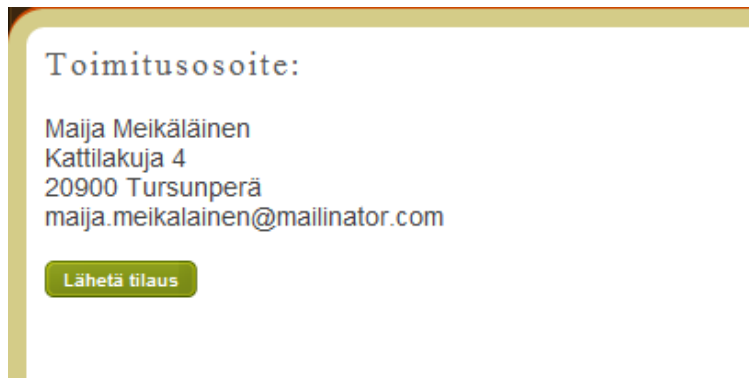
Tässä kohtaa käyttäjällä on mahdollisuus vielä ennen tilaamista tarkistaa ostoskorinsa sisällön, päivittää siinä olevien tuotteiden lukumääriä tai vaikka poistaa jonkun tuotteen korista kokonaan. Tilauksen tekemistä hän pääsee jatkamaa klikkaamalla ”Jatka tilaamista” -painiketta.

Et ole kirjautunut sisään. Voit tehdä sen nyt tai rekisteröityä uudeksi asiakkaaksi. Jos et halua rekisteröityä, voit täyttää tietosi alla olevaan lomakkeeseen.

Anna toimitusosoite...		...tai kirjaudu sisään:	
Etinimi:	Sukunimi:	Käyttäjätunnus:	
<input type="text"/>	<input type="text"/>	<input type="text"/>	
Sähköposti:		Salasana:	
<input type="text"/>		<input type="text"/>	
Katuosoite:		<input type="button" value="Kirjaudu"/> <small>Tunnukset hukassa?</small>	
<input type="text"/>			
Postinumero:	Postitoimipaikka:		
<input type="text"/>	<input type="text"/>		
<input type="button" value="Lähetä"/>			

Kuva 18. Toimitustietojen keräys.

Jos käyttäjä ei tässä vaiheessa ole sisäänkirjautuneena, annetaan hänelle mahdollisuus joko kirjautumiseen tai vaihtoehtoisesti osoitetietojensa täyttämiseen käsin.



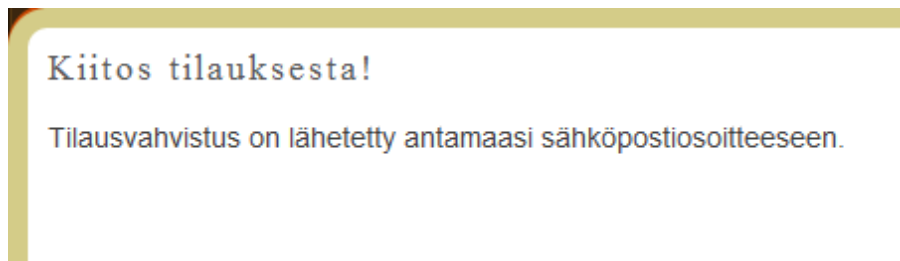
Toimitusosoite:

Maija Meikäläinen
Kattilakuja 4
20900 Tursunperä
maija.meikalainen@mailinator.com

Lähetä tilaus

Kuva 19. Toimitusosoitteen tarkistus.

Tämän jälkeen käyttäjälle annetaan vielä mahdollisuus tarkistaa osoitetietojensa oikeellisuus. Kaiken ollessa kunnossa lähetetään tilaus painamalla ”Lähetä tilaus” -nappia.



Kiitos tilauksesta!

Tilausvahvistus on lähetetty antamaasi sähköpostiosoitteeseen.

Kuva 20. Tilauksen vastaanottoilmoitus.

Tämän jälkeen ruudulle tulostuu kuvan x ilmoitus jossa käyttäjälle kerrotaan tilauksen vastaanottamisesta. Seuraavaksi lähetetään tilausvahvistus (kuva 21) käyttäjän ilmoittamaan sähköpostiosoitteeseen.

Inbox for: maija.meikalainenAlternate Address for this Inbox: MBR-0vpkby@mailinator.com ([whats this?](#))

To: **maija.meikalainen**

From: **admin@viihdevarasto.fi**

Subject: **Viihdevarasto - tilausvahvistus**

Charset: **ISO-8859-1 (view as UTF-8)**

Olemme vastaanottaneet tilauksenne:

Inland Empire - Draama

1 kpl à 8.90 €

Machinae Supremacy - Overworld

1 kpl à 12.95 €

Machinae Supremacy - A View From The End of The World

1 kpl à 16.95 €

XBOX 360 - Fable 3

2 kpl à 29.90 €

yhteensä 98.60 €

Tarkistakaa ovatko tilaamanne tuotteet oikein ylläolevassa listassa.

Ystävällisin terveisin Viihdevaston ylläpito.

Kuva 21. Tilausvahvistus.

6 TUOTEHALLINTA

Koska vähänkään suuremman tuotemäärän lisääminen tietokantaan pelkän phpMyAdminin avulla todettiin työlääksi ja melko monimutkaiseksi, päätettiin sivustolle tehdä pienimuotoinen hallinnointisivu, joka helpottaa tuotteiden lisäämistä tietokantaan.

6.1 Tuotteen lisääminen tietokantaan

Viihdevarasto Tuotehallinta

Musiikki Elokuvat Pelit

Musiikki

Esittäjä: Scar Symmetry

Nimike: Valitse tietokannasta

Levy-yhtiö: Nuclear Blast

Genre: Melodic Death Metal

Julkaisuvuosi: 2009

Formaatti: CD DVD

Hinta: 12,95

Kansikuva: scarsymmetry.jpg

Kappaleet:

- |The Iconoclast
- |The Consciousness Eaters
- |Noumenon and Phenomenon
- |Ascension Chamber
- |Mechanical Soul Cybernetics
- |Nonhuman Era
- |Dark Matter Dimensions
- |Sculpture Void
- |A Paranthesis in Eternity
- |Frequencyshifter
- |Radiant Strain

Lukumäärä:

Ohjeet

- 1: Valitse aina joko tietokannassa jo olevista tiedoista alasvetovalikosta tai kirjoita uusi lisättävä tieto viereiseen tekstikenttään.
- 2: Valitse julkaisuvuosi alasvetovalikosta.
- 3: Erottele kappaleiden nimet toisistaan | -merkillä seuraavasti:
|Kappale 1
|Kappale 2
|Kappale 3
...jne
- 4: Käytä kappaleiden nimissä heittomerkin (') sijaan aksentimerkkiä (`).

Kuva 22. Lomake musiikin lisäämisen tietokantaan tuotehallinnan kautta.

```

1 <label class="label" for="artist">Esittäjä.</label><br />
2 <select class="adminoption" name="artist" id="artist">
3   <option class="choose" value=""> Valitse tietokannasta </option>
4 <?php
5   $sql="SELECT DISTINCT all_music.title1 FROM all_music ORDER BY title1 ASC";
6   $query=mysql_query($sql);
7
8   while ($row=mysql_fetch_array($query)) {
9     $found=true;
10  ?>
11     <option value="<?php echo $row['title1'] ?>">
12       <?php echo $row['title1'] ?>
13     </option>
14 <?php
15   }
16 ?>
17 </select>

```

Koodi 9. Alasvetovalikon ”esitäyttö”.

Tuotteen tietokantaan lisääminen on toteutettu HTML-lomakkeella joka on tavallaan ”esitäytetty” eli sen alasvetovalikoihin on haettu tietokannasta esimerkiksi kaikki uniikit yhtyeiden nimet (koodi 9), eikä käyttäjän tarvitse aina uutta levyä lisätessään kirjoittaa kaikkia tietoja uudelleen. Tämä ominaisuus on toteutettu SQL:n ”SELECT DISTINCT”-komennolla (rivi 5), jolla saadaan noudettua tietokannasta kaikki uniikit tulokset. Esimerkin tapauksessa kukin esittäjä haetaan erikseen vain kerran, vaikka samalla esittäjällä olisikin monta eri julkaisua tietokannassa. Saadut yksilölliset tulokset on while-lauseessa tulostettu option-tagin value-parametrin arvoiksi (rivit 8-15).

Halutessaan käyttäjä voi toki lisätä uudenkin esittäjän kirjoittamalla sen oikean puoleiseen tekstikenttään. Sama koskee kaikkia rivejä joissa on mahdollisuus valita esitäytettyjä tietoja.

```

1 $artist = $_GET['artist'];
2 $newartist = $_GET['newArtist'];
3
4 if ($artist == "" and isset ($newartist)) {
5     $artist = $newartist;
6 }
7
8 $sql = "SELECT * FROM artist WHERE title1 = '$artist'";
9 $query = mysql_query($sql);
10 $row = mysql_fetch_array($query);
11 $num = mysql_num_rows($query);
12
13 switch ($num) {
14
15     case 0: //uusi esittäjä
16         $sql = "INSERT INTO artist (title1) VALUES ('$artist')";
17         $query = mysql_query ($sql);
18         $sql = "SELECT * FROM artist WHERE title1 = '$artist'";
19         $query = mysql_query($sql);
20         $row = mysql_fetch_array($query);
21         $artist_id = $row[id];
22         break;
23     case ($num > 0): //vanha esittäjä
24         $artist_id = $row[id];
25         break;
26 }

```

Koodi 10. Esittäjän lisäys tietokantaan.

Esimerkissä (koodi 10) on esitetty esittäjän lisääminen tietokantaan. Aluksi tallennetaan lomakkeesta saaduista tiedoista muuttujat ”\$artist” ja ”\$newartist” (rivit 1 ja 2). Muuttuja ”\$artist” tulee tietokannasta suoraan valittavan esitäytetyn alasvetovalikon

arvosta ja muuttuja "\$newartist" sen viereisestä tekstikentästä, jolla käyttäjä voi tallentaa tietokantaan kokonaan uuden esittäjän. Seuraavaksi tutkitaan kumpaa tapaa käyttäjä on käyttänyt ehtolauseen avulla (rivi 4). Eli jos muuttuja "\$artist" on tyhjä ja muuttuja "\$newartist" on asetettu on käyttäjä halunnut lisätä tietokantaan uuden esittäjän ja muuttujan "\$newartist" arvo tallennetaan muuttujaan "\$artist" (rivi 5). Seuraavaksi tehdään muuttujalla "\$artist" kysely tietokantaan tietokantaan (rivi 8) ja tallennetaan saatujen tulosten lukumäärä muuttujaan "\$num" käyttäen funktiota "mysql_num_rows" (rivi 11). Tämän muuttujan arvoa vertaillaan sitten switch – case lauseparilla. Jos arvo on nolla täytyy kyseessä olla uuden esittäjän joten nimi tallennetaan tietokannan taulun "artist" sarakkeen "title1" arvoksi (rivi 16). Koska koko tuotteen lisäämiseen tarvitaan esittäjän id-numeroa, tehdään tietokantaan uusi haku ja tallennetaan tuloksista äsken lisätyn uuden esittäjän id-numero muuttujaan "\$artist_id". Jos muuttujan "\$num" arvo on kuitenkin suurempi kuin nolla (rivi 23), tarkoittaa tämä, että esittäjä löytyy jo tietokannasta. Tällöin pitää vain tallentaa rivin kahdeksan kyselyn tuloksena saatu id-numero muuttujan "\$artist_id" arvoksi (rivi 24) myöhempää käyttöä varten.

Muut tiedot tallennetaan tarvittaessa tietokantaan sekä omiksi muuttujikseen vastaavalla tavalla, jonka jälkeen koko tuote on valmis lisättäväksi tietokantaan:

```

1 "INSERT INTO music (
2     title2,
3     released,
4     price,
5     cover,
6     tracks,
7     count,
8     date_added,
9     artist_id,
10    genre_id1,
11    format_id,
12    label_id)
13    VALUES (
14        '$title',
15        '$year',
16        '$price',
17        '$cover',
18        '$tracks',
19        '$count',
20        now(),
21        '$artist_id',
22        '$genre_id',
23        '$format_id',
24        '$label_id'
```

```
25 );
```

Koodi 11. Tuotteen lisääminen music-tauluun.

Ennen tietojen lisäämistä tehdään lomakkeesta saaduilla tiedoilla tietokantakysely. Jos kyselyn tuloksena saatujen tulosrivien lukumäärä on nolla eli tuotetta ei löytynyt aiemmin tietokannasta, tallennetaan tiedot koodin 11 mukaisesti. Jos tulosrivien lukumäärä on suurempi kuin nolla, löytyy tuote jo tietokannasta ja vain sen varastosaldo päivitetään. Vanhaan saldoon lisätään uusien tuotteiden lukumäärä:

```
1 $oldcount = $row[count];
2 $newcount = $oldcount + $count;
3 mysql_query("UPDATE music SET music.count = '$newcount' WHERE music.title2 = '$title'");
```

Koodi 12. Tuotteen varastosaldon päivitys.

Rivillä yksi tallennetaan muuttujan ”\$oldcount” arvoksi tietokantakyselyn tuloksena saatu tuotteen lukumäärä. Sen arvo lisätään käyttäjän lomakkeeseen kirjoittama varastoon lisättävien tuotteiden lukumäärään ”\$count” ja näin saatu summa tallennetaan muuttujaan ”\$newcount” (rivi 2). Rivillä kolme päivitetään tuotetta vastaavan music-taulun rivin sarakkeen ”count” arvoksi muuttujan ”\$newcount” arvo.

Tuote	Vuosi	Genre	Levy-yhtiö	Hinta	Kpl	Kansi	Formaatti	Kappaleet
Scar Symmetry - Dark Matter Dimensions	2009	Melodic Death Metal	Nuclear Blast	12.95	12		CD	01. The Iconoclast 02. The Consciousness Eaters 03. Noumenon and Phenomenon 04. Ascension Chamber 05. Mechanical Soul Cybernetics 06. Nonhuman Era 07. Dark Matter Dimensions 08. Sculpture Void 09. A Paranthesis in Eternity 10. Frequencyshifter 11. Radiant Strain

Kuva 23. Lisäämisen jälkeinen ilmoitus.

Tuotteen tietokantaan lisäämisen jälkeen tulostetaan käyttäjälle taulukko, joka sisältää lisätyn tuotteen tiedot. Jos käyttäjä ei kuitenkaan ole täyttänyt kaikkia lisäys-sivun

lomakkeen kenttiä tai hän on vaihtoehtoisesti valinnut samalta riviltä sekä esitetyt arvot että kirjoittanut uuden arvon tietokantaan lisättäväksi, tulostetaan ruudulle virheilmoitus. Jos tuote löytyi jo tietokannasta, ilmoitetaan se käyttäjälle ja tulostetaan yllä olevan taulukon lisäksi tuotteen vanha ja uusi, päivitetty varastosaldo.

7 YHTEENVETO

Työ osoittautui paljon suuritöisemmäksi kuin olin alunperin suunnitellut. Työn edetessä omat vaatimukset nousivat niin visuaalisella kuin teknisellä puolella. Eteen tuli jatkuvasti uusia haasteita ratkaistaviksi. Uusia asioita tuli myös paljon, sillä työssä käytetyt työkalut olivat minulle CSS:ää ja XHTML:ää lukuunottamatta entuudestaan tuntemattomia. Välillä projekti tuntui paisuvan kuin pullataikina jatkuvasti mieleen tulleiden uusien ideoiden ja ominaisuuksien myötä, mutta lopputulos on mielestäni aikalailla juuri sitä mitä hainkin – helppokäyttöinen, toimiva ja näyttävä nettikauppasovellus.

Koska työ toteutettiin ilman asiakasyritystä ja sen ei ollut tarkoituskaan tulla oikeasti käyttöön, ei minun tarvinnut työtä tehdessä miettiä käytännön ongelmia, jotka sivuston käytössä saattaisi ilmaantua, jos olisi niin sanotusti ”tosi kyseessä”. Miten toteutettaisiin esimerkiksi varastosaldojen reaaliaikainen toimiminen usean asiakkaan käyttäessä sovellusta samanaikaisesti, niin ettei vaikka tuotetta, jota on yksi kappale varastossa pysty laittamaan yhtäaikaista useampaan kuin yhden käyttäjän ostoskoriin ja niin edelleen? Entäpä tietoturva-asiat? Olisiko sivuston koodi turvallinen oikeassa käytössä? Pohdin näitä asioita, mutta ajan hivetessä päätin rajata ne tällä kertaa työn ulkopuolelle mahdollisen jatkokehityksen kohteiksi. Toinen jatkokehittelyä kaipaava ominaisuus on minusta rekisteröityneille käyttäjille tarjottavien palvelujen, kuten vaikka tilaushistorian selaaminen tai tallennettavan ostoskorin lisääminen. Tietokannan hallintaan kehittelemäni tuotehallinta-sivut kaipaavat myös lisää ominaisuuksia, vaikka ne nykyisessä yksinkertaisessa muodossaan ovat oiva apu tuotteiden tietokantaan lisäämiseen. Ainakin usean tuotteen yhtäaikaista lisääminen ja mahdollisuus tuotteiden poistoon olisivat hyviä lisäominaisuuksia tulevaisuuden varalle. Lisäksi tarvittaisiin

varmasti jonkinlainen tilaustenhallintasovellus, josta näkyisi tehdyt tilaukset ja niiden tila.

Mielestäni aihevalinta oli mielenkiintoinen ja sopi opinnäytetyön aiheeksi hyvin. Opin työtä tehdessäni paljon PHP:stä ja MySQL:stä ja jos nyt aloittaisin tekemään samaa työtä uudelleen, saattaisi lopputulos olla ainakin koodillisesti melkoisen eri näköinen. Koska en ennen ollut toteuttanut läheskään tämän luokan projektia, tuli koodistani melkoinen viidakko, jonka uumenista olisi ollut vaikea löytää etsimäänsä ilman Google Chrome -selaimen vakiovarusteena olevaa ”Inspect element” -ominaisuutta, jonka avulla sain kätevästi selville aina esimerkiksi kulloinkin mietinnän alla olevaan elementtiin vaikuttavat CSS-muotoilut.

LÄHTEET

Gasston, P. 2011. The Book of CSS3 – A Developer's Guide to The Future of Web Design. San Francisco: No Starch Press.

Hovi, A. Huotari, J. Lahdenmäki, T. 2005. Tietokantojen suunnittelu ja indeksointi. Jyväskylä: Docendo.

jQuery. Viitattu 07.04.2011. <http://www.jquery.com>

Linjama, T. 2001. XHTML. Jyväskylä: Docendo.

MySQL :: About MySQL. Viitattu 18.05.2011. <http://www.mysql.com/about/>

PHP-opas – mureakuha. Viitattu 27.05.2011. <http://wiki.mureakuha.com/wiki/PHP-opas>

PHP: What is PHP? Viitattu 07.04.2011. <http://www.php.net/manual/en/intro-what-is.php>

Saarikumpu, O. 1997. HTML:n ja XHTML:n väliset erot. Viitattu 28.02.2012. <http://weppipakki.com/html/tekstit/x-html.htm>

Saarikumpu, O. 2010. JavaScript-kielen alkeet – osa 1. Viitattu 07.04.2011. <http://www.weppipakki.com/js/opas/alkeet1.htm>

Schafer, S. M. 2010. HTML, XHTML and CSS Bible, 5th Edition. Indianapolis: Wiley Publishing Inc.

Sulkko, Samuli. 2011. Html5 – mikä se on ja mitä se tarkoittaa?. Viitattu 12.12.2011 <http://www.tietoviikko.fi/msareena/msblogit/bittiviidakko/html5+ndash+mika+se+on+ja+mita+se+tarkoittaa/a688822>