

SATAKUNNAN AMMATTIKORKEAKOULU



Antti-Pekka Tammisto

2007

VERKKOKAUPPASOVELLUKSEN ARKKITEHTUURI JA TOTEUTUS

Tekniikka Rauma  
Tietotekniikan koulutusohjelma

## VERKKOKAUPPASOVELLUKSEN ARKKITEHTUURI JA TOTEUTUS

Tammisto, Antti-Pekka

Satakunnan ammattikorkeakoulu

Tekniikka Rauma

Tietotekniikan koulutusohjelma

Yritys: Fin 158 Oy

Joulukuu 2007

Ohjaaja: yliopettaja, Yrjö Auramo

UDK-luokka: 004.65

Asiasanat: ohjelmointi, PHP, tietokannat, verkkokauppa

Tässä opinnäytetyössä toteutettiin verkkokauppasovellus ja sen hallintajärjestelmä. Verkkokauppa päätettiin toteuttaa itse valmiiden kaupallisten verkkokauppasovellusten sijaan, jotta sovelluksesta saataisiin mahdollisimman hyvin hallittava ja sopiva yrityksen tarpeet huomioon ottaen. Tämän projektin tavoitteena oli tehostaa yrityksen tuotteiden myyntiä sekä näkyvyyttä Internetissä.

Web-sivustolla toimivan hallintajärjestelmän tarkoitus oli helpottaa tilausten sekä verkkokaupassa näytettävien tuotteiden hallintaa. Hallintasovelluksen kautta ylläpitäjä voi kätevästi lisätä, muokata tai poistaa tuotteita ja tuotetietoja ilman monimutkaisten tietokantakäskyjen opettelua.

Kustannustehokkaista syistä projektia varten perustettiin oma web-palvelin, jonka alustaksi valittiin Debian GNU/Linux -käyttöjärjestelmä. Web-palvelinohjelmaksi valittiin ilmainen avoimeen lähdekoodiin perustuva Apache HTTP Server. Verkkokaupan sekä hallintajärjestelmän toiminnallisuus toteutettiin PHP-ohjelmointikielellä. Tietokantana käytettiin MySQL-tietokantaa, jonne luotiin tarvittavat taulut asiakkaita, tuotteita sekä tilauksia varten. Verkkokauppasivuston rakenne toteutettiin pitkälti XHTML-merkintäkielellä sekä sivujen ulkoasun muotoilu CSS-tyyliohjejärjestelmällä. Lisäksi sivujen toiminnallisiin osiin käytettiin JavaScriptiä.

## ARCHITECTURE AND IMPLEMENTATION OF E-COMMERCE APPLICATION

Tammisto, Antti-Pekka

Satakunta University of Applied Sciences

School of Technology Rauma

Information Technology

Commissioned by Fin 158 Oy

December 2007

Tutor: Yrjö Auramo, Principal Lecturer

UDC: 004.65

Keywords: programming, PHP, database, e-commerce

In this final year project a web-based e-commerce application and appropriate management system was created. The e-commerce was developed by the writer, instead of using commercial e-commerce applications, to meet the requirements of the client. This also allowed personal modification and management of the e-commerce. The purpose of this project was to improve the sales of the products and the visibility of the company on the Internet.

The web-based management system was created to facilitate the management of the orders and products of the e-commerce. With the management system the administrator can easily add, modify or remove products and product information without using complicated database commands.

For the financial reasons, the project was built to the dedicated web server with the Debian GNU/Linux platform. The web server application was chosen to be a free open source based product, Apache HTTP Server. The functionality of the e-commerce and the management system was implemented with the PHP programming language. The required information about clients, products and orders was stored in the MySQL database. The structure of the e-commerce and management system was created mostly with the XHTML markup language. The CSS stylesheet language was used to design the web page layout. In addition, some of the dynamic parts of the e-commerce and management system were created with JavaScript.

# SISÄLLYS

## TIIVISTELMÄ

## ABSTRACT

1	JOHDANTO .....	5
2	OHJELMOINTIKIELET JA OHJELMISTOT .....	6
2.1	Asiakaspuolen rakenne .....	7
2.1.1	(X)HTML .....	7
2.1.2	CSS .....	7
2.1.3	JavaScript .....	9
2.1.4	DHTML .....	9
2.2	Palvelinpuolen rakenne .....	10
2.2.1	PHP .....	10
2.2.2	Evästeet ja istunnot .....	11
2.2.3	MySQL .....	12
2.2.4	Apache HTTP-palvelimena.....	13
3	RELAATIOTIETOKANNAN TOTEUTUS .....	14
3.1	Yleistä relaatiotietokannoista .....	14
3.2	Relaatiotietokannan suunnittelu .....	14
3.3	Relaatiotietokannan toteutus .....	16
4	VERKKOKAUPPASOVELLUKSEN TOTEUTUS.....	18
4.1	Sivuston rakenne .....	18
4.2	Sivuston toiminta .....	20
4.3	Tuotteiden listaus .....	22
4.4	Tuotteen esittelysivu .....	23
4.5	Ostoskorin toteutus .....	25
4.6	Ilmoitus tuotteen lisäyksestä ostoskoriin .....	27
4.7	Ostoskorin muokkaus .....	29
4.8	Tilautustietojen oikeellisuuden varmistus .....	30
4.9	Tilausvahvistuksen lähetys .....	32
4.10	Web-tilastointi.....	34
5	HALLINTASOVELLUKSEN TOTEUTUS .....	35
5.1	Sivuston toiminta .....	35
5.2	Sivuston rakenne .....	37
5.3	Järjestelmään kirjautuminen.....	38
5.4	Tuote-sivu .....	40
5.5	Tuotteen esittelysivun luonti .....	41
5.6	Tietojen lisäys ja muokkaus .....	44
5.7	Tietojen poisto.....	47
6	YHTEENVETO .....	48
	LÄHDELUETTELO .....	49

# 1 JOHDANTO

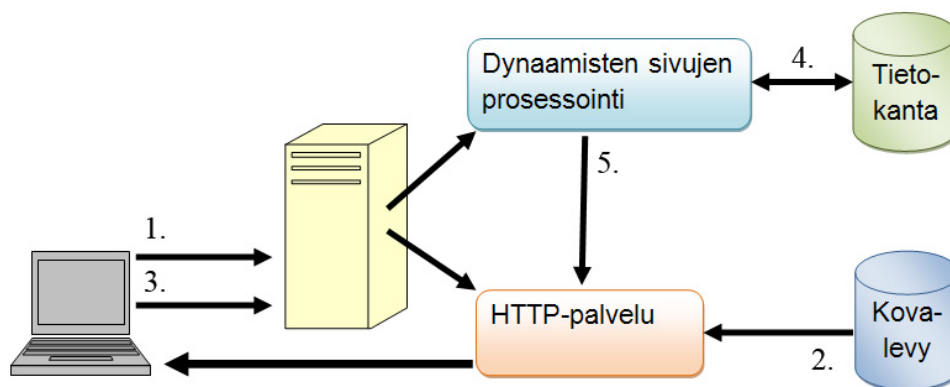
Verkkokauppojen suosio on viime vuosien aikana kasvanut nopeasti. Ostaminen verkkokaupasta on vaivatonta, koska asiakas voi tilata tuotteita omalta päätteeltään ajasta ja paikasta riippumatta. Hyvin toimiva verkkokauppa säästää aikaa ja vaivaa myös myyjältä. Verkkokaupan hallintasovelluksen kautta myyjä voi helposti lisätä ja muokata verkkokaupassa myytäviä tuotteita. Kaikki verkkokaupan kautta tehdyt tilaukset voidaan lähettää automaattisesti myyjän sähköpostiin ja lisätä tietokantaan myöhempää tarkastelua varten. Verkkokauppa toimii tyypillisesti vähintään valtakunnan laajuisesti, jolloin tavoitettavissa on myös suuri asiakaskunta.

Tässä projektissa toteutettiin Internetissä toimiva verkkokauppasovellus, jonka tarkoituksena on lisätä Fin 158 Oy:n tuotteiden myyntiä ja näkyvyyttä. Fin 158 Oy:llä ei ollut aikaisemmin web-sivuja, joten verkkokauppasivusto koettiin tarpeelliseksi. Kaupallisten verkkokauppasovellusten sijaan projekti päätettiin toteuttaa itse omalle web-palvelimelle, jotta verkkokaupasta saataisiin yrityksen tarpeita vastaava ja helposti muokattava. Verkkokaupalle hankittiin oma verkkotunnus, joka ohjattiin verkkokaupan web-palvelimelle. Verkkokauppasovelluksen ylläpitäjälle toteutettiin hallintasovellus, jonka kautta voi muun muassa lisätä ja muokata verkkokaupassa näytettäviä tuotteita sekä hallita tilauksia. Verkkokauppasivuston käytön seuranta varten web-palvelimelle asennettiin tilastointisovellus, jonka avulla ylläpitäjä voi seurata muun muassa verkkokaupan kävijämääriä ja suosituimpia sivuja.

Verkkokauppasovellusta varten luotiin oma MySQL-tietokanta, johon lisättiin tarvittavat taulut asiakkaille, tuotteille ja tilauksille. Pääasiallisena ohjelmointikielenä verkkokauppa-, ja hallintasovelluksen toteutuksessa käytettiin PHP-ohjelmointikieltä. PHP sisältää tietokantojen käsittelyyn tarvittavat funktiot ja se soveltuu erinomaisesti dynaamisten web-sivustojen tuottamiseen. Verkkokaupan ja hallintasivuston rakenne toteutettiin XHTML-merkintäkielellä ja ulkoasun muotoiluun käytettiin css-tyyliohjejärjestelmää.

## 2 OHJELMOINTIKIELET JA OHJELMISTOT

Internetin toiminta perustuu asiakas-palvelin-malliin. Webissä asiakasohjelmana toimii tyypillisesti web-selain, joka lähettää palvelimelle pyyntöjä käyttäjän ohjaamana. Web-palvelin vastaa asiakasohjelman palvelupyyntöihin esimerkiksi lähettämällä selaimelle web-sivun katseltavaksi. Ohjelmakoodi voidaan suorittaa joko asiakkaan koneella tai palvelimella. (Rantala 2005, 2.) Seuraavassa kuvassa on hahmotettu asiakas-palvelin-mallin toimintaa lomake-esimerkillä.



Kuva 1. Dynaamisten sivujen käsittely asiakas-palvelin-mallissa.

1. Käyttäjän selainohjelma pyytää web-palvelimelta lomakesivun.
2. Web-palvelin lähettää sivun palvelinohjelman kautta käyttäjän selaimelle.
3. Käyttäjä täyttää ja lähettää lomakkeen tiedot web-palvelimella suoritettavalle ohjelmalle (esim. php-sovellus).
4. Lomakkeen tiedot käsitellään ohjelmassa. Ohjelmassa voidaan esimerkiksi
  - tarkistaa syötteiden oikeellisuus (esim. onko tarvittavat lomakekentät täytetty ja käytetty sallittuja merkkejä)
  - tallentaa tiedot tietokantaan.
5. Ohjelma palauttaa käyttäjälle tietoja esimerkiksi varmistuksen tietojen lähetyksen onnistumisesta. (Ekonoja, Lahtonen & Mäntylä 2007.)

Tässä opinnäytetyössä esitellyt ohjelmointitekniikat on eritelty asiakaspuolen ja palvelinpuolen ohjelmointitekniikoihin, joita käsitellään tarkemmin seuraavissa alaluvuissa.

## 2.1 Asiakaspuolen rakenne

Asiakaspuolen ohjelmointi käsittää perinteisesti staattisten web-sivujen tuottamisen. Staattisilla sivuilla tarkoitetaan käytännössä sitä, että sivujen muuttaminen vaatii teki-jänsä toimia esimerkiksi web-dokumentin julkaisemisen aina muutosten jälkeen. (Ran-tala 2005, 3.)

Staattiset sivustot on tyypillisesti toteutettu HTML-pohjaisella merkintäkielellä. Jotta web-sivuihin saataisiin dynaamista toiminnallisuutta, voidaan HTML:n ohella käyttää erilaisia skriptikieliä kuten JavaScriptiä. (Wikipedia 2006: JavaScript.)

### 2.1.1 (X)HTML

HTML (HyperText Markup Language) on WWW:n (World Wide Web) hypertekstido-kumenttien esittämiseen tarkoitettu merkintäkieli. XHTML (Extensible HyperText Markup Language) on edelleen kehitetty versio HTML:sta, joka määrittää HTML-semantiikan tarkasti XML-merkkauksen ehdoilla toisin kuin epätarkempi HTML. (Pel-tomäki & Nykänen 2006, 9-10.)

Verkkokaupan ja hallintajärjestelmän web-sivuston rakenne toteutettiin pitkälti XHTML-merkintäkielellä. Sivut koodattiin itse graafisten HTML-editorien sijaan, jotta koodista saisi selkeää ja myös myöhemmin ymmärrettävää. WYSIWYG-tyyppiset ko-tisivueditorit lisäävät usein omia merkintöjään koodiin, mikä tekee osaltaan koodista sekavampaa. Jotkut kotisivueditorit saattavat kirjoittaa virheellistä HTML-koodia, jol-loin sivut eivät välttämättä näy oikein kaikilla web-selaimilla.

### 2.1.2 CSS

CSS (Cascading Style Sheets) on merkintäjärjestelmä, jolla voidaan määrittää web-sivujen ulkoasu (Korpela 2003, 2). CSS-tyyli voidaan yhdistää haluttuun HTML-dokumenttiin periaatteessa kolmella eri tavalla. Yksi tapa on kirjoittaa tyylisääntö suo-raan haluttuun HTML-elementtiin style-parametrilla, esimerkiksi `<p style="color: red">... </p>`. Tämä on kuitenkin hankala tapa, koska tyylimäärittely täytyy kirjoittaa jokai-seen haluttuun elementtiin. Samasta syystä dokumentin päivittämisestä tulee myös han-kalaa. Toinen tapa on upottaa tyylisääntö HTML-dokumentin `<head>`-osaan `<style>`-elementillä. Tämä tyyli toimii vain niissä dokumenteissa, joissa tämä määrittely on tehty.

Kolmas tapa on käyttää ulkoista CSS-tyylitiedostoa, jota käytettiin myös tässä työssä ulkoasun muotoiluun. Ulkoisen tyylitiedoston käyttäminen on suositeltavaa, kun halutaan säilyttää koko sivuston ulkoasu yhtenäisenä pienellä vaivalla. Tyylitiedostoon voidaan määrittellä jokaiselle web-elementille omat yksilölliset ulkoasumäärittelyt. Seuraavassa esimerkissä css-tyylitiedostoon on määritelty leipätekstin (p) ja pääotsikon (h1) tyyli-ominaisuudet sekä sivun taustakuva (Koodi 1).

```
01  p {
02      font-family: "Trebuchet MS"; font-size: 80%;
03  }
04
05  h1 {
06      font-family: Verdana; font-size: 120%; color: #ffffff;
07      text-decoration: bold; text-align: center;
08  }
09
10  body {
11      background-image: url('tausta.jpg')
12  }
```

Koodi 1. Esimerkki CSS-tyylitiedoston sisällöstä.

Esimerkissä leipätekstin (p) ja pääotsikon (h1) kirjainlaji on asetettu font-family-ominaisuudella, jonka jälkeen tulee arvo eli kirjainlajin nimi. Fontin koko on vastavasti määritetty font-size-ominaisuudella ja arvo prosentteina suhteessa ympäröivän elementin fonttikokoon. Sivujen taustakuvaksi on <body>-elementin background-image-ominaisuudella määritelty tausta.jpg-kuvatiedosto.

Käytännössä ulkoisen CSS-tyylitiedoston vaikutukset nähdään kun luodaan HTML-dokumentti, joka sisältää CSS-tyylitiedoston määrittelemiä elementtejä. HTML-dokumentissa pitää olla myös viittaus tyylitiedostoon. Tällöin sivujen elementtien ulkoasua voidaan päivittää tekemällä muutokset pelkästään CSS-tiedostoon. Ulkoinen tyylitiedosto saadaan käyttöön viittaamalla siihen jokaisella web-sivulla, jonka ulkoasu halutaan määrittää tyylitiedoston ohjeistuksen mukaisesti. Tyylitiedostoon viittaus kirjoitetaan HTML-dokumentin lähdekoodiin <head> ja </head>-tagien väliin. Seuraavassa esimerkki ulkoisen CSS-tyylitiedoston liittamisestä HTML-dokumenttiin (Koodi 2).



```

01 <html>
02 <head>
03 <title>Sivun otsikko</title>
04 <link rel="stylesheet" type="text/css" href="tyyli.css" />
05 </head>
06 <body>
07   Sivun varsinainen sisältö
08 </body>
09 </html>

```

Koodi 2. Css-tyylitiedoston liittäminen HTML-dokumenttiin.

Koodi 2:n esimerkissä css-tiedostoon viittaus on tehty <link>-elementillä, jonka tyyppi on annettu type-attribuutilla ”text/css” (rivi 4). Tyylitiedoston (tyyli.css) sijainti on määritelty href-attribuutilla.

### 2.1.3 JavaScript

JavaScript on yleisin www-dokumenteissa käytetty asiakaspuolen skriptikieli. JavaScript-ohjelmakoodi kirjoitetaan script-elementin sisään, ja se sijaitsee tavallisesti HTML-dokumentin alussa dokumentin otsikko-osassa <head> ja </head> -tagien välissä. Se voidaan sijoittaa myös dokumentin runko-osaan <body> ja </body> -tagien väliin tai käyttää kokonaan erillistä JavaScript-tiedostoa, johon viitataan HTML-dokumentissa. (Tuikka 2006.) Seuraavassa yksinkertainen esimerkki JavaScriptin käytöstä (Koodi 3).

```

01 <html>
02 <body>
03
04 <script type="text/javascript">
05   document.write("Hei, maailma!")
06 </script>
07
08 </body>
09 </html>

```

Koodi 3. JavaScript upotettuna HTML-dokumentin runko-osaan.

### 2.1.4 DHTML

Nykyään vuorovaikutteisten web-sivujen toteutustekniikasta puhutaan usein nimellä DHTML (Dynamic HTML). DHTML on termi, jota käytetään kuvaamaan HTML:n, tyylisäännösten ja skriptien yhdistelmää, joka mahdollistaa dokumenttien elävöittämisen. Tyypillisesti DHTML-tyyppisissä web-sivuissa on käytetty (X)HTML-merkintäkieltä, CSS-tyyliohjejärjestelmää sekä JavaScriptiä. Kyse ei ole kuitenkaan HTML-kielen laajennuksesta vaan eri tekniikoiden yhdistelmästä. (Korpela 2006.)

## 2.2 Palvelinpuolen rakenne

Vaikka asiakaspuolen tekniikoilla voidaan saavuttaa tiettyä dynaamisuutta web-sovellukseen, dynaamisuus rajoittuu pääasiassa käyttöliittymän hallintaan. Puhuttaessa tietojärjestelmien hyödyntämisestä webissä tarvitaan lähes poikkeuksetta palvelinpuolen tekniikkaa. Perinteisesti tietojärjestelminä käytetään relaatiotietokantoja, joiden käsittely perustuu SQL-kieleen. Tyypillisesti web-sovellukset joudutaan sisällyttämään näihin tietokantoihin. Tällaisen toteutuksen pohjalle tarvitaan kokonaisvaltainen palvelinratkaisu, joka koostuu esimerkiksi käyttöjärjestelmä-, web-palvelin- ja tietokantapalvelimesta.

Verkkokauppasovelluksen sekä hallintajärjestelmän toiminnallisuus on toteutettu suurimmaksi osaksi palvelinpuolen tekniikoilla. Näistä tekniikoista lisää seuraavissa kohdissa.

### 2.2.1 PHP

PHP (PHP: Hypertext Preprocessor) on palvelinpuolen ohjelmointikieli, jota käytetään etenkin web-palvelinympäristössä dynaamisten web-dokumenttien luomiseen. Ensimmäiset PHP:n versiot julkaistiin 1995. Nykyisin PHP on vertailuissa johtava dynaamisten web-palveluiden tuottamiseen tarkoitettu kieli. (<http://fi.wikipedia.org/wiki/PHP>.)

PHP-koodi tulkitaan aina ohjelman suoritusvaiheessa web-palvelimella kääntämisen sijaan. Tämä helpottaa ohjelman testailua ja nopeuttaa sen muuttamista. PHP-koodi kirjoitetaan yleensä `<?php` ja `?>` tai `<? ja ?>` -tagien väliin, joista web-palvelin tunnistaa suoritettavan PHP-koodin. (Wikipedia 2006: PHP.)

Asiakaspuolella tagien väliin kirjoitettua PHP-koodia ei voi nähdä esimerkiksi web-sivun lähdekoodista, koska PHP-koodi suoritetaan aina palvelimella eikä sitä lähetetä takaisin asiakaspuolelle. Seuraavassa esimerkki PHP-koodista (Koodi 4).

```
01 <?php
02 echo "Hei, maailma!";
03 ?>
```

Koodi 4. Esimerkki PHP-koodista, joka tulostaa tekstin "Hei, maailma!"

### 2.2.2 Evästeet ja istunnot

Evästeillä tarkoitetaan web-palvelimen HTTP-otsakkeena selaimelle lähettämää merkkijonoa, joka on tarkoitettu tallennettavaksi selaimen muistiin tai levyille. Evästeet mahdollistavat HTTP-protokollasta puuttuvan mekanismin web-sovellusten tilatiedon ylläpitämiseen HTTP-pyyntöjen ja -vastausten välillä. HTTP-pyyntöissä toistuvasti lähetettävän evästeen avulla tiettyyn istuntoon kohdistuvat pyynnöt voidaan tunnistaa. Istunnolla tarkoitetaan tässä yhteydessä sellaista kokoelmaa HTTP-pyyntöjä ja -vastauksia, joille voidaan määritellä aloitus- ja lopetustapahtuma. (Rantala 2005, 218.)

Tässä työssä evästeitä käytettiin ostoskorin tuotteiden ja niiden määrien säilyttämiseen istunnon aikana. PHP:ssa evästeet asetetaan setcookie()-funktiolla (Koodi 5). Setcookie()-funktiota tulee kutsua aina ennen varsinaisen HTML-koodin alkamista PHP-sivulla.

```
01 <?php
02 setcookie("Auto", "Toyota", time() + 60 * 2)
03 ?>
```

Koodi 5. Evästeen asetus PHP:ssa.

Setcookie()-funktion ensimmäinen parametri on evästeen nimi, jolla siihen voidaan myöhemmin viitata. Toisena parametrina on evästeen arvo eli tallennettava tieto ja kolmantena evästeen voimassaoloaika. Jos evästeelle ei määritetä voimassaoloaikaa, niin se pysyy voimassa selaimen sulkemiseen asti. Edellisessä esimerkissä evästeen nimeksi on määritetty Auto, arvoksi Toyota ja voimassaoloajaksi kaksi minuuttia. Istunnot toimivat PHP:ssa siten, että palvelun käyttäjälle muodostetaan oma tunnistee, joka tallennetaan evästeeseen tai siirretään URL:n mukana. Palvelimelle luodaan PHP:n asetusten määrittämään kansioon tunnistetta vastaava tiedosto, johon istunnossa käytetyt muuttujat tallentuvat.

### 2.2.3 MySQL

MySQL on hyvin yleisesti web-sovelluksissa käytetty SQL-relaatiotietokannan hallintajärjestelmä. MySQL:n hallinnointi tapahtuu tavallisesti komentoriviltä tai tekstipohjaisella asiakasohjelmalla. (Wikipedia 2006: MySQL.) Tässä työssä MySQL-tietokannan päälle rakennettava ohjelmalogiikka toteutettiin PHP -ohjelmointikielellä, joka tarjoaa tarvittavat funktiot MySQL-tietokannan käsittelyyn.

Tietokantapalvelimeen saadaan muodostettua yhteys `mysql_connect()`-funktiolla. Funktion parametreiksi määritetään palvelimen nimi, käyttäjätunnus ja salasana. Parametrien puuttuessa yhteys yritetään muodostaa omalle palvelimelle (localhost) ilman salasanaa. Kun yhteys tietokantapalvelimeen on luotu, valitaan käytettävä tietokanta `mysql_select_db()`-funktiolla. Funktion parametrit ovat tietokannan nimi ja yhteyden tunniste. (Ohjelmointiputka 2007.) Seuraavassa esimerkki yhteydenmuodostuksesta tietokantapalvelimeen ja tietokannan valinnasta.

```
01 <?php
02 $dbhost="127.0.0.1";
03 $dbuser="mattim";
04 $dbpass="abc123";
05 $dbname="fin158";
06
07 mysql_connect($dbhost, $dbuser, $dbpass) or die("Yhteysvirhe");
08 mysql_select_db($dbname) or die("Virheellinen tietokanta");
09 ?>
```

Koodi 6. Yhteydenmuodostus tietokantapalvelimeen PHP:n `mysql_connect()` -funktiolla sekä tietokannan valinta `mysql_select_db()` -funktiolla.

Rivillä 7 oleva kohta `"or die("Yhteysvirhe")"` keskeyttää yhteydenmuodostuksen ja tulostaa virheilmoituksen käyttäjälle, jos tietokantaan ei saada yhteyttä.

MySQL-tietokanta luotiin omalle web-palvelimelle verkkokaupan tuotteita, tilauksia ja asiakkaita varten. Samaa tietokantaa hyödyntää myös verkkokaupan hallintasovellus.

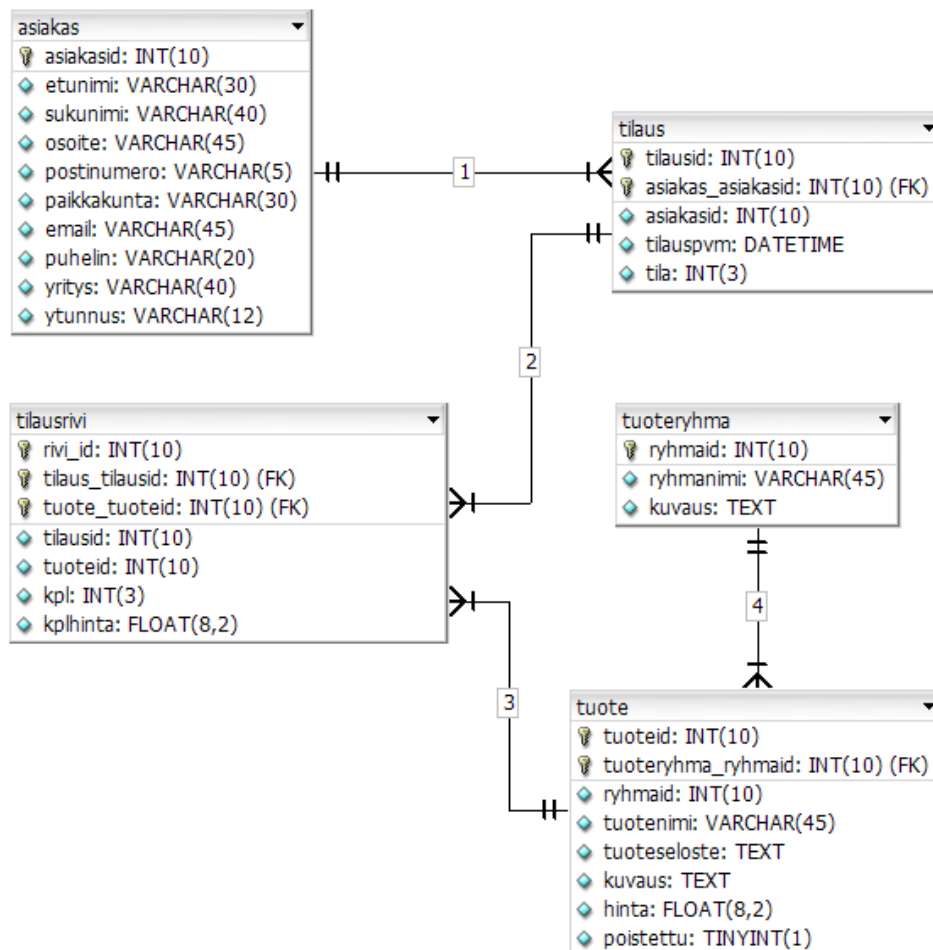
#### 2.2.4 Apache HTTP-palvelimena

Web-palvelimessa täytyy aina olla palvelinohjelmisto, joka mahdollistaa muun muassa web-sivujen lataamisen ja tietokantasovellusten käytön web-sivuilla palvelimen ja asiakkaan välillä (Sundström 1998). Palvelinohjelmistoksi valittiin ilmainen, avoimeen lähdekoodiin perustuva HTTP-palvelinohjelma Apache HTTP Server. Apache tukee staattisten tiedostojen jakamista HTTP-protokollan yli. Apachen HTTP-palvelin on saatavilla Unixille (BSD:t, Linux), Windowsille sekä monelle muulle käyttöjärjestelmälle. Apache HTTP Server on vuoden 1996 huhtikuusta lähtien ollut Internetin suosituin HTTP-palvelinohjelmisto. (Wikipedia 2007: Apache.)

Tässä työssä Apache HTTP-palvelimen alustana käytettiin Debian GNU/Linux -käyttöjärjestelmää.



sen käsittelyn tila kuten ”käsittelyssä” tai ”toimitettu”. Tilausrivi-tauluun kirjataan tarvittavat tiedot tilatuista tuotteista ja tehdään tarvittavat viittaukset muihin tauluihin. Tilausrivi-taulun viiteavaimeksi on määritelty tuoteid-kenttä, jonka perusteella tiedetään mihin tuote-taulun tuotteeseen viitataan. Toiseksi viiteavaimeksi on määritelty tilaus-taulun tilausid, jolla tilaus-taulun ja tilausrivi-taulun väliset tiedot saadaan pidettyä ehyenä. Lisäksi tilausrivi-taulu sisältää kpl-kentän, johon tallennetaan asiakkaan tilaamien tuotteiden kappalemäärä. Tilausrivi-taulun kplhinta-kenttään tuotteelle voidaan määrittellä tuote-taulun hinta-kentästä poikkeava hinta jos esimerkiksi asiakkaalle on annettu alennusta usean tuotteen ostamisesta. Asiakas-tauluun kirjataan tuotteiden tilauksen yhteydessä asiakkaan yhteystiedot. Tuotetaulujen sisältö tietotyyppineen on tarkemmin näkyvissä seuraavassa kuvassa. Viiteavain on merkitty tietokentän perään (FK) -merkinnällä (”foreign key”).



Kuva 2. Tietokannan rakenne ja yhteydet taulujen välillä.

Tietokannan taulujen relaatiot kuvassa ovat seuraavat:

1. Asiakas voi tehdä yhden tai useamman tilauksen.  
Tilauksen voi tehdä vain yksi asiakas.
2. Tilaukseen voi kuulua yksi tai useampi tilausrivi.  
Tilausrivi voi kuulua vain yhteen tilaukseen.
3. Tilausrivillä voi olla vain yksi tuote.  
Tuote voi olla yhdellä tai useammalla tilausrivillä.
4. Tuote voi kuulua vain yhteen tuoteryhmään.  
Yhteen tuoteryhmään voi kuulua monta tuotetta.

### 3.3 Relaatiotietokannan toteutus

Verkkokaupan ja hallintasovelluksen yhteisesti käyttämä relaatiotietokanta luotiin käsin syöttämällä tarvittavat komennot MySQL:n komentokehoteessa. Ensimmäinen vaihe oli luoda uusi fin158-niminen tietokanta komennolla ”CREATE DATABASE fin158;”. Tämän jälkeen luotiin edellisessä kappaleessa mainitut taulut tietojen tallentamista varten. Seuraavassa esimerkki tuote-taulun luonnista.

```

01 CREATE TABLE `tuote` (
02   `tuoteid` int(10) NOT NULL auto_increment,
03   `ryhmaid` int(10) unsigned NOT NULL,
04   `tuotenimi` varchar(45) NOT NULL,
05   `tuoteseloste` text NOT NULL,
06   `kuvaus` text NOT NULL,
07   `hinta` float(8,2) default NULL,
08   PRIMARY KEY (`tuoteid`)
09   FOREIGN KEY (`ryhmaid`)
10   REFERENCES tuoteryhma(`ryhmaid`)
11   ON DELETE NO ACTION
12   ON UPDATE CASCADE
13 )
14 TYPE=INNODB;

```

Koodi 7. Tuote-taulun luonti MySQL:n komentokehoteessa.

Tuote-nimisen taulun luonti tapahtuu ”CREATE TABLE” -komennolla, jonka jälkeen sulkujen sisälle määritetään tietokentät tietotyyppeineen. ”NOT NULL” tarkoittaa, ettei kenttään hyväksytä tyhjiä arvoja. Auto\_increment tuoteid-kentän kohdalla tarkoittaa, että tuoteid:n arvoa kasvatetaan automaattisesti yhdellä, kun esimerkiksi tuotteita lisätään tuote-tauluun. Perusavaimeksi tuote-tauluun on ”PRIMARY KEY” -komennolla määritelty tuoteid-kenttä. Viiteavaimeksi on ”FOREIGN KEY” -komennolla määritelty ryhmaid-kenttä, joka viittaa tuoteryhma-taulun perusavaimeen. Itse viittaus tehdään



”REFERENCES”-komennolla, jonka perään kirjoitetaan viitattava taulu ja sulkujen sisälle viitattavan taulun perusavain. ”ON DELETE NO ACTION” -komento estää tuoteryhmän poistamisen tuoteryhma-aulusta, mikäli tuote-aulusta löytyy kyseiseen tuoteryhmään kuuluvia tuotteita. Tuote-aulun ryhmäid-kenttä on siis sidottu tuoteryhma-aulun perusavaimeen, jolloin taulujen välillä on viite-eheys. Tuoteryhmän poistaminen onnistuu käytännössä vain poistamalla ensin tuoteryhmään kuuluvat tuotteet tuote-aulusta ja sen jälkeen tuoteryhmä tuoteryhma-aulusta. Rivin 12 komento ”ON UPDATE CASCADE” päivittää vastaavasti ryhmäid-kentän arvot tuote-, ja tuoteryhma-auluun, jos arvoja muutetaan. Viite-eheyden toimivuuden edellytyksenä on, että tietokantatyypinä käytetään InnoDB:tä. Tietokantatyypä voidaan määritellä rivin 14 mukaisesti komennolla ”TYPE=INNODB”.

Muiden taulujen luonti tapahtuu vastaavalla tavalla kuin Koodi 7:n esimerkissä. Periaatteessa vain taulun nimi ja kentät tietotyyppeineen vaihtelevat taulukohtaisesti. Kun taulu on luotu, voidaan niihin tallentaa verkkokaupan kautta tehdyt tilaustiedot. Kirjoittaminen tauluihin tapahtuu PHP:n tietokannankäsittely-funktioilla, jotka on sisällytetty verkkokaupan toimintaan. Tauluihin tallennettuja tietoja voidaan muokata myös hallintasovelluksen kautta. Näihin palataan vielä tarkemmin seuraavissa luvuissa.

## 4 VERKKOKAUPPASOVELLUKSEN TOTEUTUS

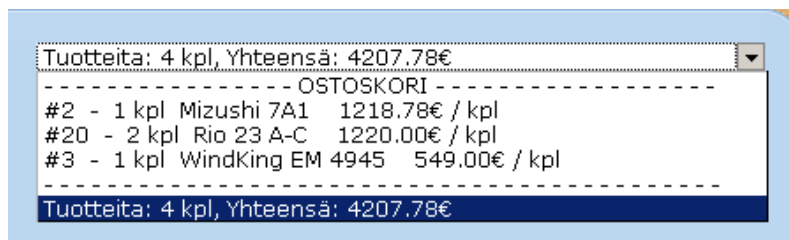
### 4.1 Sivuston rakenne

Verkkokauppasivuston visuaalinen rakenne toteutettiin taulukoimalla. Taulukoiden avulla tekstit, kuvat ja grafiikat voidaan sijoittaa haluttuun kohtaan. Taulukon leveys ja korkeus määriteltiin prosentteina, jolloin sen koko mukautuu näytön kuvatarkkuuden mukaan. Tällä tavalla sivusto saadaan näkymään paremmin halutulla tavalla eri kuvatarkkuuksilla. Taulukoiden reunaviivat piilotettiin ulkoasuyistä muilta osin paitsi navigointivalikosta. Navigointivalikko sijoitettiin oikealle omaan taulukkoon, josta käyttäjä voi klikkaamalla avata haluamansa sivun. Navigointivalikkoon luotiin linkit tuoteryhmien sivuille sekä etu-, info-, yhteystiedot-, palaute-, linkit- ja ostoskorisivulle. Tuoteryhmien nimet haetaan tietokannan tuoteryhmä-tilusta navigointivalikkoon, johon voi kätevästi luoda uusia tuoteryhmiä verkkokaupan hallintasoveluksen kautta. Tuoteryhmä-linkit hakevat tietokannasta valitun tuoteryhmän tuotteet, jotka tulostetaan verkkokaupan sivulle omaan taulukkoon. Kaikki verkkokaupan sivut toteutettiin PHP-tiedostoina, koska suuri osa sivuista sisältää PHP-koodia. Seuraavassa kuvassa verkkokaupan etusivu (Kuva 3).



Kuva 3. Näkymä verkkokaupan etusivusta.

Verkkokaupan miniostoskori sijoitettiin sivuston oikeaan yläreunaan alavetolistaan, jota klikkaamalla asiakas pääsee aina näkemään lisäämänsä tuotteet. Ostoskorin lisääminen alavetolistaan tuntui sopivalta, koska se vie vain vähän tilaa sivulta ja sen voi avata aina tarvittaessa. Ostoskorissa näytetään tuotteiden id, kappalemäärä, nimi, hinta sekä loppuyhteenvetona tuotteiden yhteismäärä ja yhteishinta (Kuva 4).



Kuva 4. Verkkokaupan ostoskori alavetolistassa sivun oikeassa yläkulmassa.

Verkkokaupan muuttumattomien sivuelementtien sisällyttämiseksi sivuille käytettiin PHP:n include-komentoa. Include-komennolla voidaan web-sivun mihin tahansa kohtaan liittää ennaltamääritetyn tiedoston sisältämät tiedot. Verkkokaupan sivustossa esimerkiksi verkkokaupan logo, valikko ja ostoskori ovat sellaisia elementtejä, jotka on hyvä lisätä include-komennolla sivuille. Näin voidaan helpottaa huomattavasti sivuston ylläpitoa, koska tulevat muutokset tarvitsee tehdä vain include-tiedostoon, josta ne tulevat voimaan kaikille sivuille. Jos esimerkiksi verkkokaupan navigointivalikkoon lisätäisiin uusi tuoteryhmä, pitäisi muutos tehdä erikseen jokaisen sivun valikkoon. Sama koskee kaikkia sivuja missä ei ole käytetty yhtenäistä sisällytettyä tiedostoa.

Verkkokauppasivustolla käytettiin pääosin kahta eri include-tiedostoa, header.inc ja footer.inc. Header.inc tiedostoon on lisätty kaikki sivun alkupään vakiotoiminnot kuten yhteyden avaaminen tietokantaan, alkuosan sivupohja, ostoskori, HTML:n meta-elementit, css-tiedostot sekä yleisesti käytetyt javascriptit. Footer.inc-tiedosto sisältää muun muassa navigointivalikon sekä loppuosan sivupohjasta. Headerin ja footerin väliin toteutetaan itse sivun ominaissisältö, joka voi olla esimerkiksi yhteystiedot tai tietyn tuoteryhmän tuotteet tulostettuna. Seuraavassa esimerkki tiedostojen sisällyttämisestä HTML-koodin sekaan PHP:n include-komennolla (Koodi 8).

```
01 <html>
02 <head>
03 <title>Fin 158 - etusivu</title>
04 <?php include_once("header.inc"); ?>
05 <table border="0" align="center" cellpadding="4" cellspacing="0">
06 <tr>
07 <td colspan="5" height="50" align="center" valign="middle">
08 <h4>Tervetuloa Fin 158 Oy - verkkokauppaan!</h4>
09 </td>
10 </tr>
11 </table>
12 <?php include_once("footer.inc"); ?>
```

Koodi 8. Tiedostojen sisällyttäminen HTML-koodiin PHP:n include-komennon avulla.

Riveillä 4 ja 12 käytetty `include_once()`-funktio varmistaa, ettei samaa tiedostoa sisällytetä kuin kerran. Tällä on haluttu varmistaa se, ettei esimerkiksi CSS-tyylitiedostot tule vahingossa lisätyksi kahteen kertaan samalle sivulle.






Sivustosta pyrittiin tekemään toimiva, näyttävä ja helppokäyttöinen. Huomiota kiinnitettiin muun muassa sivuston keveyteen sekä resoluutioriippuvuuteen eli siihen miten sivut näkyvät ja toimivat eri näytön kuvatarkkuuksilla. Sivuston toiminnassa otettiin huomioon selainriippuvaisuus eli se, miten sivut toimivat eri selaimilla. Sivujen testaamiseen käytettiin kolmea suosittua selainta: Internet Exploreria, Mozilla Firefoxia ja Operaa.

Verkkokauppasivuston rakenteen toteuttamiseen käytettiin XHTML-kieltä ja ulkoasun muotoiluun CSS-tyyliohjejärjestelmää. Sivuston toiminnalliset osat kuten ostoskorin toiminta ja tuotteiden listaus toteutettiin pitkälti PHP-kielellä. JavaScriptiä käytettiin muun muassa tulostamaan asiakkaalle ilmoitus tuotteen lisäämisestä ostoskoriin sekä varmistamaan ostoskorin tyhjentämistoiminto.

## 4.2 Sivuston toiminta

Asiakas voi selata verkkokaupan tuotteita valitsemalla sivuston navigointivalikosta tietyn tuoteryhmän. Navigointivalikon tuoteryhmä-linkit suorittavat tietokantakyselyn MySQL-kannasta, jolloin tietyn tuoteryhmän tuotteet tulostetaan taulukkoon asiakkaan nähtäväksi. Taulukkoon tulostetaan tuotteen id-numero, nimi, lyhyt tuoteseloste ja hinta. Jokaisen tuoterivin loppuun tulostetaan linkki tuotteen esittelysivun avaamista varten sekä painike, josta tuotteen voi lisätä ostoskoriin. Esittelysivulla näytetään tarkemmat

tiedot tuotteesta sekä tuotteen kuva. Seuraavassa kuvassa näkymä tuotetaulukosta verkkokaupan sivustolla.

Vesiautomaatit					
# ↓	Tuotenimi	Tuoteseloste	Hinta	Tiedot	Lisää
4	Aqua Filter FDL	Vesijohtoverkkoon liitettävä lattiamallinen vesiautomaatti. Helppokäyttöinen ja samalla tyylikäs raikasvesiautomaatti, joka lisää viihtyisyyttä missä vaan. Vesiautomaatista saa raikasta, jäädytettyä (4 – 12 C°) sekä kuumaa vettä (85 – 100 C°), josta saa vaivattomasti lämpimiä juomia. Lisäksi vielä huoneenlämpöistä vettä (n. 21 C°). Käyttökohteina taukotiilat, neuvottelutilat ja toimistot. Aqua Filter on vesijohtoverkkoon liitettävä vesiautomaatti.	1342.00	<a href="#">Näytä</a>	
5	Aqua Filter FDS	Vesijohtoverkkoon liitettävä pöytämallinen vesiautomaatti Helppokäyttöinen ja samalla tyylikäs raikasvesiautomaatti. Toimii vesianostelijana jäädytetylle, kuumennetulle ja huoneenlämpöiselle vedelle. Lisävarusteena saatavana erilaisia tiskin alle sijoitettavia suodatinvaihtoehtoja. Erinomainen ratkaisu ruokaloiden palvelulinjoille. Vesijohtoverkkoon liitettävä vesiautomaatti.	939.40	<a href="#">Näytä</a>	
19	Connect UV 23 A	Vesijohtoverkkoon liitettävä lattiamallinen vesiautomaatti Huoltohälytysjärjestelmällä ja annostelualueen UV valolla Laite soveltuu kohteisiin, joissa vaaditaan erityistä puhtautta.	1464.00	<a href="#">Näytä</a>	
20	Rio 23 A-C	Vesijohtoverkkoon liitettävä lattiamallinen vesiautomaatti paikkaan missä ihmiset kaipaavat virkistystä, jossa vedenkulutus on suurta.	1220.00	<a href="#">Näytä</a>	
21	Rio Jetset	Vesijohtoverkkoon liitettävä pöytämallinen vesiautomaatti. Tiskin päällä näyttävä vesiautomaatti kahviloihin ja ruokaloihin.	671.00	<a href="#">Näytä</a>	

Kuva 5. Tuotetaulukko verkkokaupan sivustolla.

Asiakas voi lisätä tuotteita ostoskoriin klikkaamalla ostoskorin kuvaketta tuotteen vierestä. Klikkaamalla useamman kerran ostoskorin kuvaketta lisätään tuotteen kappalemäärää aina yhdellä ostoskorissa. Ostoskori-sivulla asiakas voi poistaa tuotteita, muuttaa tuotteiden kappalemääriä tai tyhjentää koko ostoskorin. Ostoskori-sivulta asiakas pääsee myös jatkamaan tilauksen tekoa tilauslomakkeeseen. Tilauslomakkeessa asiakkaalta kysytään toimitus-, ja laskutustiedot sekä yritysasiakkailta yrityksen nimi ja vapaavalintaisesti y-tunnus. Kun tilauslomakkeen tiedot on tarkistettu, lähetetään tilausvahvistus sähköpostitse sekä asiakkaalle että myyjälle. Samalla tiedot tilauksesta tallennetaan tietokantaan.

Asiakkaan tekemien toimintojen varmistaminen toteutettiin JavaScriptin confirm()-funktioilla. Confirm()-funktio avaa käyttäjän selaimen pienen ponnahdusikkunan, jossa on esitetty kysymys vastausvaihtoehtoina ok tai peruuta. Tällaisia varmistusta vaativia toimintoja ovat muun muassa ostoskorin tyhjennys sekä hallintasivustolla tuotteiden ja tilausten poistaminen.

### 4.3 Tuotteiden listaus

Tuotteet haetaan tietokannasta verkkokaupan sivuille seuraavalla MySQL-kyselyllä.

```
01 $query = mysql_query("SELECT * FROM tuote WHERE ryhmäid=3 AND
02 poistettu=0");
```

Koodi 9. Tuotteiden haku tietokannasta tietyn tuoteryhmän mukaan.

Edellä oleva kysely hakee tuote-aulusta kaikki tuotteet, joiden ryhmäid on 3 ja joiden poistettu-kentän arvo on 0. Tässä tapauksessa ryhmäid 3 viittaa tuoteryhmään ”vesiautomaatit”. Poistettu-kentän arvon ollessa 0 tuote näytetään verkkokaupassa ja arvon ollessa 1, tuotetta ei näytetä. Tuotetiedot tulostetaan taulukkoon, joka muodostetaan tuotteiden rivimäärien mukaan seuraavalla tavalla.

```
01 for($laskuri = 0; $laskuri < mysql_numrows($query); $laskuri++) {
02     $id = mysql_result($query,$laskuri,"tuoteid"); ?>
03     <tr><td><p>
04     <?echo mysql_result($query,$laskuri,"tuotenimi")?></p></td>
05
06     <td><a href=<?echo 'tuote.php?showid='.$id.'?'>
07     onClick="window.open(this.href,'pop'); return false;"
08     target="newWin">Näytä</a></td>
09
10     <?echo '<td><a href="?addid='.$id.'" name="lisaa">
11     </a></td>';
12     echo '</tr>';
13 }
```

Koodi 10. Tuotetaulukon muodostus tuoterivien lukumäärän mukaan.

MySQL-kyselyllä haettujen tuotteiden määrä voidaan palauttaa mysql\_numrows-komennolla (rivi 1). Tuotetaulukko muodostetaan for-silmukan sisällä luomalla tarvittavat taulukkosolut sekä tulostamalla tuotetiedot omille riveilleen. For-silmukkaa käydään läpi kunnes saavutetaan MySQL-kyselyn tuottama tuoterivien määrä. Rivillä 4 oleva mysql\_result()-komento palauttaa tuotteen nimen laskurin mukaisesta kohdasta. Muut tuotetiedot saadaan haettua taulukkoon samalla tavalla. Joka tuoterivin perään tulostetaan lisäksi linkit tuotteen esittelysivun avaamista sekä tuotteen ostoskoriin lisäämistä varten. Kumpikin linkki käyttää GET-metodia tuotteen id-numeron välittämiseen palvelimelle. Esimerkiksi rivillä 6 oleva linkki esittelysivun avaamiseen 'tuote.php?showid='.\$id.'" välittää tuotteen id-numeron ”showid”-nimisessä GET-muuttujassa. Esittelysivulla tuotteen tiedot osataan hakea juuri tämän välitetyn id:n perusteella. For-silmukan yhteydessä tuotekohtaiset id-numerot tallennetaan \$id-

muuttujaan, johon viitataan linkeissä. Tällä tavalla viittaus saadaan kohdistettua id-numeroa vastaavaan tuotteeseen esimerkiksi jos käyttäjä avaa tietyn tuotteen esittelysivun auki. Samalla periaatteella toimii myös linkki ”<a href=?addid=’.\$id.’>”, josta tuote voidaan lisätä ostoskoriin. Tässä tapauksessa GET[’addid’]-muuttuja käsitellään ostoskorin toiminnasta vastaavassa cart.php-tiedostossa, jossa suoritetaan toimenpiteet tuotteen lisäämiseksi ostoskoriin. Tästä lisää kohdassa ”4.5 Ostoskorin toteutus”.

Tuotetaulukon tietojen järjestämiseen käytettiin Stuart Langridgen ilmaista Sorttable-JavaScriptiä. Langridgen kehittämä JavaScript on ilmainen kaupalliseen käyttöön, joten se soveltui hyvin käytettäväksi sellaisenaan tässä projektissa. Tuotteet saadaan järjestettyä esimerkiksi hinnan mukaan nousevasti tai laskevasti klikkaamalla tuotetaulukon hintasarakkeen otsikkoa. JavaScriptillä toteutettuna järjestäminen on tietoturvan kannalta turvallista, koska järjestämistapaa ei tarvitse lähettää palvelimelle \$\_GET-metodilla. Järjestäminen voidaan toteuttaa MySQL-tietokantajärjestelmän ”ORDER BY”-komennolla, mutta tässä tapauksessa järjestämistavan välittämiseen joudutaan usein käyttämään \$\_GET-metodia. Suuren tietomäärän järjestäminen rasittaa myös tietokantaa, koska järjestäminen joudutaan käsittelemään aina tietokannassa. Järjestämistä vaativa taulukko joudutaan myös lataamaan aina uudelleen asiakkaan selaimeen, jos uudelleenjärjestely suoritetaan MySQL-kyselyllä. JavaScriptillä taulukon uudelleenjärjestely voidaan toteuttaa ilman sivun uudelleenlataamista. (Langridge 2007.)

#### 4.4 Tuotteen esittelysivu

Tuotteiden esittelysivuna käytetään tuote.php-tiedostoa, johon tuotteen tiedot kuvaa lukuun ottamatta haetaan tietokannasta. Tuote.php on esivalmisteltu sivu, johon on määriteltä valmis taulukkorakenne tuotetiedoille. Tuotetiedot haetaan taulukon tiettyihin soluihin tuotteen id:n perusteella aina siitä tuotteesta, jonka kohdalta esittelysivu avataan. Tällä menetelmällä jokaiselle verkkokaupan tuotteelle ei tarvitse luoda omaa esittelysivua, vaan tuotetiedot saadaan kätevästi haettua tietokannasta tuotekohtaisesti tuote.php-sivulle. Esittelysivulla tuotekuva näytetään keskitettynä, tuotetiedot näkyvät siirtävällä pohjalla ja tekstit sekä linkit ovat valmiiksi muotoiltu sopiviksi. Esittelysivu avataan näkyviin aina yhdessä samassa ponnahdusikkunassa, jolloin vältetään useilta samanaikaisilta ponnahdusikkunoilta. Seuraavassa kuvassa erään tuotteen esittelysivu (Kuva 6).

**Argo Ulisse 13 CL Free**

ARGO Ulisse Free Tehokas viilentäjä suureenkin tilaan, toimistoon, liikkeeseen, neuvotteluhuoneeseen, atk-tilaan, työtilaan, varastoon...

Laitetta ohjataan joko käyttöpaneelista tai kaukosäätimellä. Laitteessa on erillinen ulos asennettava lauhdutin joka yhdistetään viilentäjäan taipuisan putkiston välityksellä. Putken läpivienti voidaan tehdä joko seinään tai ikkunan-/ oven karmiin.

- \* Kaukosäätöohjaus
- \* Jäähdytysteho 3,7 kW
- \* Tehon tarve 1300 W
- \* Ilmamäärä max. 450 m3/h
- \* Puhallinnopeudet 3
- \* Välietkun pituus 210 cm (lisävaruste jatkoletkut 200 tai 400 cm)
- \* Mitat K/L/S: 790/580/245mm sisäyksikkö
- \* 490/525/250mm ulkoyksikkö
- \* Paino 44 kg sisäyksikkö
- \* 13 kg ulkoyksikkö

[Lataa esite \(pdf\)](#)

**Hinta:** 1909.30 € (sis. alv 22% ja toimituskulut)

Kuva 6. Tuotteen tiedot haetaan esittelysivulle (tuote.php) tietokannasta tuotteen id:n perusteella.

Esittelysivulle tulostetaan tuotteen kuva, nimi, tuotekuvaus mukaan lukien mahdolliset linkit sekä hinta. Kuva sekä tarkempi tuotekuvaus voidaan lisätä esittelysivulle verkkokaupan hallintasovelluksen kautta. Kuva tallennetaan web-palvelimelle ja nimetään tuotteen id:n perusteella (esim. 3.jpg). Seuraavassa esimerkki, miten tuotteen kuva sekä tuotekuvaus haetaan esittelysivulle MySQL-kyselyllä.

```
01 $query = mysql_query("select tuoteid from tuote where tuoteid =
02 ".$_GET['showid'].");
03 $tuoteid = mysql_result($query,"tuoteid");
04
05 $query = mysql_query("select kuvaus from tuote where tuoteid =
06 ".$_GET['showid'].");
07 $kuvaus = mysql_result($query,"kuvaus");
```

Koodi 11. Tuotteen tietojen haku tietokannasta esittelysivulle.

Koodin alussa mysql\_query-komennolla haetaan tuote-taulusta sen tuotteen id-numero, jonka kohdalta käyttäjä avaa esittelysivun auki (rivi 1 ja 2). Tuotteen id-numero tallennetaan \$tuoteid-muuttujaan, koska sitä tarvitaan myöhemmin tuotteen kuvan hakemi-



seen web-palvelimelta (rivi 3). Rivillä 5 ja 6 tuote-aulusta haetaan tuotteen kuvaus, joka tallennetaan \$kuvaus-muuttujaan. Tuotteen nimi ja hinta haetaan tuote-aulusta vastaavalla tavalla ja tallennetaan muuttujiin \$tuotenimi sekä \$hinta. Tiedot tulostetaan esittelysivulle seuraavan esimerkin mukaisesti.

```
01 <table>
02   <tr><td><?echo '';?></td></tr>
03   <tr><td><?php echo $tuotenimi;?>
04     <?php echo $kuvaus;
05     echo '<br/><b>Hinta:</b> ' . $hinta. ' €</p>';?>
06   </td></tr>
07 </table>
```

Koodi 12. Tuotetietojen tulostus muuttujista esittelysivulle.

Esittelysivun luonnista on kerrottu tarkemmin kohdassa ”5.5 Tuotteen esittelysivun luonti”.

#### 4.5 Ostoskorin toteutus

Ostoskorin toteutus oli tärkeä, mutta myös haastava vaihe verkkokauppasovelluksen tekemisessä. Virtuaalinen ostoskori toimii verkkokaupassa tilattavien tuotteiden ja niiden kappalemäärien väliaikaisena säilytyspaikkana. Käytännössä ostoskoriin lisätyt tuotteet säilytetään aina tilaukseen asti, josta tilaustiedot tallennetaan tietokantaan myöhempää tarkastelua varten. Ostoskori on välttämätön osa verkkokauppasovellusta, koska asiakkaan tilaamat tuotteet täytyy pystyä tallentamaan istunnon ajaksi.

Ostoskorin toiminta toteutettiin PHP:lla käyttäen hyödyksi evästeitä. Evästeet osoittautuivat tähän käyttötarkoitukseen sopiviksi, koska lisätyt tuotteet kappalemäärineen voidaan tallentaa selaimen muistiin koko istunnon ajaksi. Evästeitä käyttämällä välttyään myös tallentamasta tietokantaan ostoskorin sisältämiä väliaikaisia tuotetietoja. Evästeitä käytettäessä ostoskorin toiminnallisuus edellyttää, että käyttäjän selain myös hyväksyy evästeiden asettamisen. Suosituimmissa selaimissa evästeiden asettaminen on kuitenkin oletusarvoisesti sallittu, ellei käyttäjä sitä erikseen ole selaimen asetuksista estänyt. Seuraavassa esimerkki ostoskorin toteutuksesta (Koodi 9.).

```

01 <?php
02 if (isset($_GET['addid'])) {
03     $id = $_GET['addid'];
04     if (!$_COOKIE['ostoskori']) {
05         $ostoskori="$id=1";
06     }
07
08     else {
09         $ostoskori="";
10         $tilaukset=preg_split("/:/", $_COOKIE[ostoskori]);
11         $tilausolemassa=false;
12         for ($i=0;$i<count($tilaukset);$i++) {
13             $tilausrivi=preg_split("/=/", $tilaukset[$i]);
14
15             if ($tilausrivi[0]==$id) {
16                 $tilausolemassa=true;
17                 $tilausrivi[1]++;
18                 $tilaukset[$i]="$tilausrivi[0]=$tilausrivi[1]";
19             }
20
21             if ($i==0)
22                 {$ostoskori=$tilaukset[$i];}
23
24             else
25                 {$ostoskori.=":".$tilaukset[$i];}
26         }
27
28         if ($tilausolemassa==false)
29             {$ostoskori.=":$id=1";}
30     }
31     $_COOKIE[ostoskori]=$ostoskori;
32     setcookie("ostoskori", $ostoskori);
33 }
34 ?>

```

Koodi 13. Ostoskorin toteutus PHP:lla ja evästeillä.

Tässä työssä toteutetun ostoskorin toimintaidea on tallentaa käyttäjän lisäämien tuotteiden id-numerot sekä kappalemäärät evästemuuttujaan. Tuotteiden id:n perusteella kaikki tarvittavat tuotetiedot voidaan hakea tietokannasta, joten muita tietoja id:n ja kappalemäärien lisäksi ostoskoriin ei ole välttämätöntä tallentaa. Tuotteiden id-numerot ja kappalemäärät tallennetaan evästemuuttujaan ”=” -merkillä erotettuna. Ostoskorin sisältäessä useita tuotteita, erotetaan ne toisistaan ”:” -merkillä. Esimerkiksi evästemuuttujan sisältö ”1=2:8=1” tarkoittaa, että tuotetta, jonka id on 1, on lisätty kaksi kappaletta ja vastaavasti tuotetta, jonka id on 8, on lisätty yksi kappale. Evästemuuttujan sisältö pilkotaan osiin PHP:n `preg_split()`-funktiolla (rivi 10 ja 13), jolloin tuotteiden id-numerot ja kappalemäärät luetaan samalla omiin taulukkoalkioihinsa. `preg_split()`-funktio pilkkoo evästemuuttujan sisällön tietyn merkkijonossa esiintyvän merkin perusteella, joka tässä tapauksessa on ”=” -merkki sekä ”:” -merkki. Kun tuotteiden id-numerot sekä

kappalemäärät ovat omissa taulukkoalkioissaan, voidaan muutokset kohdistaa tuotetta vastaavan taulukkomuuttujan alkioon. Esimerkiksi lisättäessä ostoskorissa jo olevaa tuotetta yhdellä, osaa koodi tarkistaa tilausrivin olemassaolon (rivi 15) ja lisätä kappalemäärää tuotetta vastaavan \$tilausrivi[1] -taulukkomuuttujan alkioon (rivi 17). Ostoskoriin lisättyjen tuotteiden tiedot näytetään käyttäjälle muun muassa ostoskori-sivulla sekä sähköpostitse asiakkaalle lähetettävässä tilausvahvistuksessa.

#### 4.6 Ilmoitus tuotteen lisäyksestä ostoskoriin

Asiakkaan lisätessä tuotteita ostoskoriin, tulostetaan sivulle ilmoitus tapahtumasta. Jos ilmoitusta ei tulostettaisi, asiakas ei välttämättä huomaisi tuotteen lisäystä tapahtuneen. Seuraavassa koodilistauksessa JavaScript -toteutus ilmoituksen tulostuksesta asiakkaalle (Koodi 14).

```

01 <script language="JavaScript">
02 <!--
03   var timer=null;
04   function changeText(text) {
05     if (timer!=null) clearInterval(timer);
06     document.getElementById("valittu").innerHTML=text;
07     timer=setInterval('returnText()', 6000);
08     return false;
09   }
10   function returnText() {
11     clearInterval(timer);
12     timer=null;
13     document.getElementById("valittu").innerHTML="<br/>";
14   }
15 -->
16 </script>

```

Koodi 14. Ilmoituksen tulostus asiakkaalle tuotteen lisäämisestä ostoskoriin JavaScriptillä toteutettuna.

Koodi 14:ssä on luotu kaksi funktiota, joista `changeText(text)` tulostaa ilmoituksen näkyviin kuudeksi sekunniksi ja `returnText()` korvaa ilmoituksen rivinvaihdolla. Ilmoituksen tulostus on kohdassa `document.getElementById("valittu").innerHTML=text`. Tämä JavaScriptin funktio osaa hakea HTML-elementin sille määritetyn id:n perusteella ja sijoittaa tässä tapauksessa "text" -argumentin arvon haetun elementin tagien väliin. Ilmoitus tulostetaan asiakkaalle ostoskorin alapuolella sijaitsevaan `<div>`-elementtiin, jonka id:ksi on määritetty "valittu". Kuva 7:ssä näkymä ilmoituksen tulostumisesta.



Kuva 7. Ilmoitus tulostetaan ostoskorin alapuolelle kuuden sekunnin ajaksi.

Kun ilmoitus on tulostettu, käynnistetään samalla ajastin kohdassa ”timer=setInterval(’returnText()’, 6000)”. Ajastimen aika on määritetty millisekuntein eli ilmoitus pysyy kuusi sekuntia näkyvillä, jonka jälkeen kutsutaan returnText()-funktioita. Tämä funktio nolaa ajastimen ja tulostaa ilmoituksen tilalle rivinvaihdon. Myös changeText()-funktion alussa ajastin nolataan, mikäli se on aiemmin käynnistetty. Tämä nollaus tehdään käytännössä silloin, jos ajastimeen määritetyn kuuden sekunnin kuluessa teksti halutaan korvata uudella tekstillä. Tällainen tilanne on mahdollinen, jos asiakas lisää uuden tuotteen ostoskoriin kuuden sekunnin kuluessa edellisen tuotteen lisäämisestä. Mikäli ajastinta ei nollattaisi, uusi teksti katoaisi jo ennen kuin se on ollut näkyvissä halutut kuusi sekuntia. Jotta ilmoitus saadaan vielä käytännössä näkymään, täytyy changeText()-funktioita kutsua HTML-dokumentin runko-osassa (Koodi 15).

```

01 <?php
02 if($uusi) {
03     $query=mysql_query("SELECT nimi FROM tuote where id=".$uusi."");
04     $nimi=mysql_result($query, "nimi");?>
05     <body onload="return changeText('<?echo'<p>' . $nimi?>lisätty
06     ostoskoriin!</p>');"><?
07 }
08 else {
09     echo '<body>';
10 }

```

Koodi 15. ChangeText()-funktioita kutsuminen HTML-dokumentin runko-osassa.

ChangeText()-funktioita kutsutaan aina sivun lataamisen yhteydessä jos \$uusi-muuttuja on tosi. \$uusi-muuttujaan tallennetaan aina lisätyn tuotteen id-numero, joten sen arvo on aina tosi kun asiakas lisää uuden tuotteen ostoskoriin. Lisätyn tuotteen nimi haetaan tietokantakyselyllä \$nimi-muuttujaan \$uusi-muuttujan sisältämän id:n perusteella (rivi 3-4). Rivin 5 kohta <body onload> palauttaa changeText()-funktion aina ehdon toteutuessa. Muussa tapauksessa tulostetaan sivun runkotagi <body> ja loppusivu ladataan normaalisti ilman ilmoitusta.

#### 4.7 Ostoskorin muokkaus

Ostoskori-sivulla asiakas voi muokata ostoskorinsa sisältöä. Ostoskorin sisältämät tuotteet tulostetaan ostoskori-sivulla omaan taulukkoon, josta voi muuttaa tuotteiden kappalemääriä, poistaa tuotteita tai tyhjentää koko ostoskorin sisältö. Tilauksen jatkaminen tapahtuu ostoskori-sivulla olevasta ”Tilaa tuotteet” -painikkeesta. Seuraavassa kuvassa verkkokaupan ostoskori-sivu.

#	Kpl	Tuotenimi	Tuoteseloste	Hinta	Yht.	Tiedot
3	<input type="text" value="2"/>	WindKing EM 4945	Ilmanjäähdytin, tuuletin, kostutin, ilmanpuhdistin ja lämmitin. Kaikki yhdessä tehokkaassa paketissa jota on helppo siirrellä paikasta toiseen.	549.00€	1098.00€	<a href="#">Näytä</a>
6	<input type="text" value="1"/>	Linda	Pieni kooltaan, suuri teholtaan. Kesäkuumalla oivallinen makuuhuonekumppani tai mainio työkaveri.	725.90€	725.90€	<a href="#">Näytä</a>
21	<input type="text" value="1"/>	Rio Jetset	Vesijohtoverkkoon liitettävä pöytämallinen vesiautomaatti. Tiskin päällä näyttävä vesiautomaatti kahviloihin ja ruokaloihin.	671.00€	671.00€	<a href="#">Näytä</a>
<b>Tuotteita: 4 kpl</b> <b>Yhteensä: 2494.90 €</b>			<b>Poistaaksesi tuotteita ostoskorista,</b> <b>asetta tuotteen kplmääräksi 0 ja klikkaa 'Päivitä muutokset'</b>		<input type="button" value="Tyhjennä ostoskori"/>	
<input type="button" value="Päivitä muutokset"/>					<input type="button" value="Tilaa tuotteet"/>	

Kuva 8. Verkkokaupan ostoskori-sivu.

Asiakas voi muuttaa tuotteiden kappalemääriä ostoskorissa kirjoittamalla uuden kappalemäärän ostoskori-taulukon ”Kpl”-sarakeeseen. Jos asiakas haluaa poistaa yksittäisen tuotteen ostoskorista, pitää tuotteen kappalemääräksi asettaa 0. Muutokset täytyy päivittää ”Päivitä muutokset” -painikkeella, joka käsittelee ostoskorin sisällön uudelleen ja lataa ostoskori-sivun päivitettyä asiakkaan selaimen. Ostoskorin tyhjentäminen tapahtuu ”Tyhjennä ostoskori” -painikkeella. Seuraavassa koodilistauksessa ostoskorin tyhjentämisen toteutus.

```

01 if($_POST[tyhjenna]) {
02     if($_COOKIE['ostoskori']) {
03         $_COOKIE['ostoskori'] = "";
04         setcookie("ostoskori", "");
05     }
06 }

```

Koodi 16. Ostoskorin tyhjentäminen.

Koodin alussa tarkistetaan onko ”Tyhjennä ostoskori” -painiketta painettu. Rivillä 2 varmistetaan myös, että ostoskori-eväste on olemassa. Tämän jälkeen ostoskori-evästeen arvo asetetaan tyhjäksi ja päivitetään ostoskorin sisältö ajan tasalle setcookie()-funktioilla.

Vahinkojen välttämiseksi asiakkaalta halutaan varmistus ostoskorin tyhjentämiseen. Seuraavassa ostoskorin tyhjentämistoiminnon varmistuksen toteutus JavaScriptin `confirm()`-funktiolla.

```
01 <script language="JavaScript">
02 <!--
03 function confirm_delete() {
04   var answer = confirm("Haluatko varmasti tyhjentää ostoskorin?");
05   if (answer == true) {
06     window.location="ostoskori.php";
07   }
08   else {
09     return false;
10   }
11 }
12 -->
13 </script>
```

Koodi 17. Ostoskorin tyhjentämisen varmistus JavaScriptin `confirm()` -funktiolla.

`Confirm()`-funktio avaa pienen ponnahdusikkunan käyttäjän selaimen, jossa kysytään varmistusta ostoskorin tyhjentämiseen (rivi 4). Jos kysymykseen vastataan ”ok”, välitetään ostoskorin tyhjentämiseen tarvittavat lomaketiedot PHP-tulkille, joka suorittaa ostoskorin tyhjentämisen. Tämän jälkeen ostoskori-sivu ladataan uudelleen käyttäjän selaimen `window.location` -komennolla (rivi 6). Jos käyttäjä vastaa kysymykseen ”peruuta”, ei lomaketietoja lähetetä ja skriptin suoritus keskeytetään (rivit 8-9).

Kun mahdolliset muutokset on tehty ostoskoriin voidaan tilausta jatkaa ”Tilaa tuotteet” -painikkeella. ”Tilaa tuotteet” -painike ohjaa asiakkaan tilauslomake.php-sivulle täyttämään tilaustiedot.

#### 4.8 Tilautustietojen oikeellisuuden varmistus

Asiakkaan tilauslomakkeeseen täyttämät tilaustiedot on hyvä tarkistaa mahdollisten tyhjien kenttien tai virheellisten syötteiden varalta ennen niiden lisäämistä tietokantaan. Tällä tavoin tietokanta saadaan pidettyä ehyempänä ja estetään suuri osa asiakkaan tekemistä tahattomista virheistä tilauslomaketta täytettäessä. Tilauslomakkeen tietokenttien tarkastamiseen käytettiin PHP:n `isset()`-, `strlen()`- ja `eregi()`-funktiota. Seuraavassa esimerkki tilauslomakkeen postinumero-kentän tarkastamisesta.

```

01 $errors = NULL;
02 if(isset($_POST['pnro']) && strlen($_POST['pnro']) == 5 &&
03 eregi('^[0-9]+$', ($_POST['pnro']))) {
04     $pnro_title = 'Postinumero: ';
05     $pnro = mysql_real_escape_string(stripslashes($_POST['pnro']));
06 }
07 else {
08     $errors .= '<br/>Tarkista postinumero.';
09     $pnro_title = '<font color="red">Postinumero</font>: ';
10 }

```

Koodi 18. Postinumeron tarkistus PHP:n eregi()-funktioilla tilauslomakkeen lähettämisen yhteydessä.

Rivillä 2 kohdassa ”if(isset\$\_POST[’pnro’])” tarkistetaan, onko postinumero-kenttä olemassa eli lähetetty PHP-tulkille tilauslomakkeen yhteydessä. Postinumeron pituus on tarkistettu strlen()-funktioilla. Postinumeron tulee olla täsmälleen 5 merkkiä pitkä, jotta ehto täyttyy. Eregi()-funktioilla postinumero-kentälle määritetään sallitut merkit. Tässä tapauksessa postinumero-kentän merkeiksi hyväksytään numerot 0-9. Jos kaikki ehdot täyttyvät tallennetaan tilauslomakkeesta saatu postinumero \$pnro-muuttujaan. Kaikki hyväksytyt lomakekentät tallennetaan aina PHP-muuttujiin, jotta kertaalleen kirjoitetut tietokentät voidaan tarvittaessa palauttaa tilauslomakkeeseen. Muuttujista tiedot voidaan myös tallentaa suoraan tietokantaan kun tiedot on ensin tarkastettu. Ennen kuin tilauslomakkeen tiedot tallennetaan tietokantaan ja lähetetään sähköpostilla myyjälle ja asiakkaalle pitää \$errors-muuttujan arvon olla tyhjä (rivi 1). Käytännössä \$errors-muuttujaan lisätään aina virheviesti, joka voidaan tulostaa asiakkaalle kentän puuttuessa (rivi 8). Jos kaikki tarvittavat lomakekentät on hyväksytysti täytetty, on myös \$errors-muuttuja tyhjä.

Tilauslomake tarkistetaan aina lähettämisen yhteydessä ja jos virheitä esiintyy, tilaaja palautetaan korjaamaan puutteelliset tiedot. Puutteelliset kohdat ilmoitetaan tilaajalle punaisella kentän otsikon värillä sekä lomakkeen alareunassa olevalla ilmoitustekstillä. Jo kertaalleen täytetyt tiedot säilytetään, joten tilaajan ei tarvitse täyttää muuta kuin virheelliset kohdat ja lähettää tilauslomake uudelleen. Seuraavassa kuvassa verkkokaupan tilauslomake, joka on yritetty lähettää puutteellisena. (Kuva 9).

**Tilaustiedot**

\* -merkityt kentät ovat pakollisia.

\*Etunimi:

\*Sukunimi:

\*Toimitusosoite:

\*Postinumero:

\*Postitoimipaikka:

\*Sähköpostiosoite:

\*Puhelin:

**Laskutustiedot** (Jos eri kuin toimitustiedot)

Laskutusosoite:

Postinumero:

Postitoimipaikka:

**Yritysassiakkailla**

Yrityksen nimi:

Y-tunnus:

Huomioitavaa:

Tiedoissa oli korjattavaa.  
Tarkista punaisella merkityt kohdat.

Kuva 9. Tilaustiedot tarkistetaan ennen niiden lisäämistä tietokantaan.

#### 4.9 Tilausvahvistuksen lähetyks

Kun asiakas on tilannut tuotteet, lähetetään hänelle sähköpostitse tilausvahvistus tila-  
tuista tuotteista. Vastaavasti myyjälle lähetetään asiakkaan tekemä tilaus ja tilaustiedot.  
Tilausvahvistuksen lähettämiseen käytetään PHP:n mail()-funktiota. Seuraavassa esi-  
merkissä on esitelty tilausvahvistuksen lähetyksen toteutus.

```

01 $to = "fin158@fin158.com";
02 $subject = "Tilaus ".date('d.m.Y H:i:s');
03 $headers .= 'From: Fin 158 Oy - Verkkokauppa'. "\n";
04
05 $mailcontent = "TILAUSTIEDOT\n".$setunimi." ".$sukunimi." \n"
06 ".$osoite." \n".$postinumero." ".$paikkakunta." \n"
07 ".$email." \n".$puhelin." \n\n".$yritys." \n".$ytunnus
08 ." \n\nTILATUT TUOTTEET\n".$sisalto
09 ." \n\nHUOMIOITAVAA\n".$_POST[huom];
10
11 $mailcontent = wordwrap($mailcontent, 70);
12 mail($to, $subject, $mailcontent, $headers);

```

Koodi 19. Tilauksen lähetyks myyjälle mail()-funktiolla.



Koodin alussa on määritelty \$to-muuttuja, johon määritetään sähköpostin vastaanottaja. Sähköpostin otsikko määritellään \$subject-muuttujaan ja ylimääräinen otsakekenttä, johon esimerkiksi voidaan kirjoittaa tarkemmat tiedot viestin lähettäjistä määritetään \$headers-muuttujaan. Itse viestin sisältö muodostetaan \$mailcontent-muuttujassa (rivi 5). Viestin sisällöksi on määritelty asiakkaan tilaustiedot, jotka saadaan haettua tilauslomakkeen lähetyksen yhteydessä tallennetuista muuttujista. Lisäksi sähköpostissa lähetetään tiedot asiakkaan tilaamista tuotteista, jotka on koottu \$sisalto-muuttujaan sekä tilauslomakkeessa oleva ”huomioitavaa”-kenttä, johon asiakas voi kirjoittaa lisätietoja tilausta koskien. Sähköpostin sisältö rivitetään wordwrap()-funktiolla 70 merkin pituiseksi, jotta sähköpostin asettelusta saadaan toimiva (rivi 11). Lopuksi viesti lähetetään mail()-funktiolla, johon on parametreina määritelty edellä mainitut muuttujat. Asiakkaalle lähetetty tilausvahvistus on seuraavan kuvan mukainen.

**Aihe:** Tilaus 22.11.2006 13:00:02  
**Lähettäjä:** "Fin 158 Oy - Verkkokauppa" <fin158@fin158.com>  
**Päiväys:** ke 22.11.2006 13:00  
**Vastaanottaja:** apt@tr.samk.fi  
**Prioriteetti:** Normaali  
**Asetukset:** [Näytä kaikki otsaketiedot](#) | [Näytä tulostusversio](#) | [Lataa tämä tiedostona](#)

TILAAJAN TIEDOT  
 -----  
 Matti Meikäläinen  
 Testikuja 5  
 26100 Rauma  
 matti.m@firma.fi  
 044-123 4567

Testifirma Oy  
 1234567-8

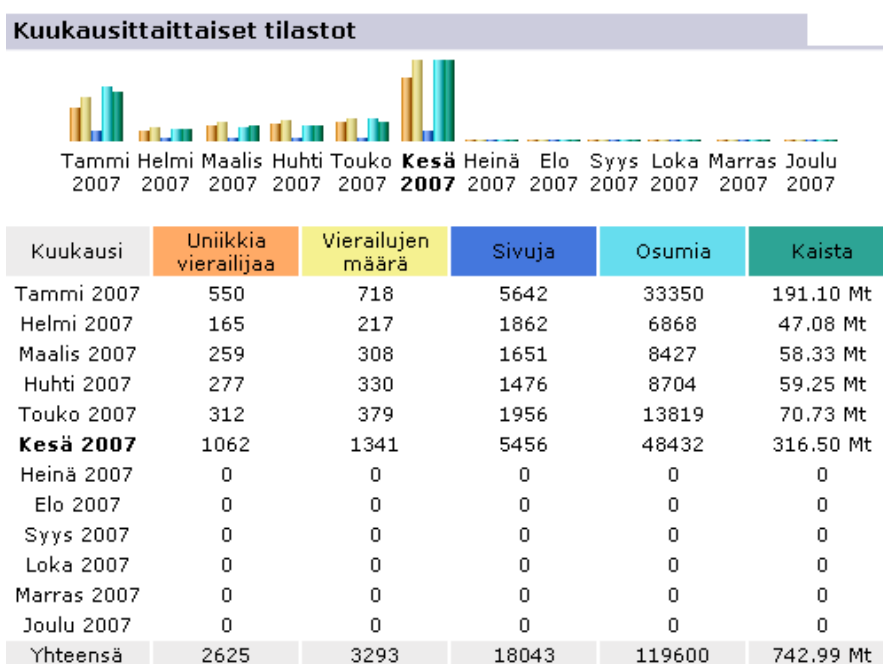
TILATUT TUOTTEET  
 -----  
 #6 1 kpl Linda 595.00€ Yht. 595€  
 #9 1 kpl Argo Ulisse 13 CL Free 1565.00€ Yht. 1565€  
 #16 7 kpl Elixair 500 856.00€ Yht. 5992€  
 #20 1 kpl Rio 23 A-C 1000.00€ Yht. 1000€

HUOMIOITAVAA  
 -----  
 Jos mahdollista toimittakaa tuotteet tämän viikon kuluessa.

Kuva 10. Tilausvahvistuksen asettelu sähköpostissa.

#### 4.10 Web-tilastointi

Verkkokauppasivuston kävijämäärien seuraamista varten palvelimelle asennettiin ilmainen AWStats-tilastointityökalu. AWStats luo webissä katseltavan tilastointisivun Apachen HTTP-palvelinohjelman loki-tiedostojen pohjalta. Tilastointisivulle on koottu muun muassa kävijämäärät päivittäin sekä 24 tunnin aikajakaumana, eniten katsotut sivut, vierailijan ip-osoite ja selain, hakusanat, joilla sivuille on saavuttu hakukoneen kautta sekä paljon muuta. AWStatsin avulla on helppo seurata verkkokaupan suosiota ja myös tutkia miten sivujen mainostaminen vaikuttaa kävijämääriin. Seuraavassa näkymä AWStatsin kuukausittaisista kävijämääristä.



Kuva 11. Näkymä AWStatsin kuukausittaisista kävijämääristä.

## 5 HALLINTASOVELLUKSEN TOTEUTUS

### 5.1 Sivuston toiminta

Verkkokaupan hallintasovelluksen tarkoitus on helpottaa verkkokauppasivuston ylläpitoa sekä tilaus-, ja tuotekannan hallintaa. Käytännössä hallintasovellus on oma käyttöliittymä tietokannan hallintaan, jonka kautta voidaan lisäksi muokata tiettyjä verkkokauppasivuston elementtejä. Ilman tämän kaltaista käyttöliittymää käyttäjän pitäisi tehdä muutokset tietokantaan aina käsin esimerkiksi konsoliliittymän kautta, joka edellyttää syvällisempää tuntemusta SQL-komentojen käyttämisestä.

Hallintasovellus toimii web-sivustolla, joten sen käyttäminen onnistuu kätevästi selaimella. Tämä kuitenkin mahdollistaa samalla hallintasovellukseen pääsyn keneltä tahansa, jonka vuoksi sivuston tietoturvaan tulee kiinnittää huomiota. Hallintasovellus suojattiin rajoittamalla pääsy sivustolle vain ylläpitäjän ip-osoitteista Apachen .htaccess-tiedoston avulla. Lisäksi sivustolle tehtiin sisäänkirjautumissivu, jonka kautta sovellusta pääsevät käyttämään vain tunnuksen ja salasanan omaavat käyttäjät.

Hallintasovelluksessa käyttäjä voi tarkastella tietokannassa olevia tauluja ja tehdä niihin kätevästi muutoksia helppokäyttöisen web-liittymän kautta. Käyttäjä voi muun muassa lisätä tuotteita myyntiin verkkokaupan sivustolle tai poistaa niitä. Hallintasovelluksessa voi myös tarkastella verkkokaupan kautta tehtyjä tilauksia. Kaikki nämä tietokantaan kohdistuvat kyselyt on koodattu valmiiksi HTML-dokumentteihin PHP:n tietokannan käsittelyfunktioilla, jolloin käyttäjä voi suorittaa kyselyt käytännössä nappeja painamalla. Hallintasovelluksessa verkkokaupan tuotteille voidaan lisäksi luoda esittelysivu, johon voidaan ladata kuvia tuotteesta ja lisätä tarkempia tuotetietoja.

Hallintasovelluksen tiettyjä toimintoja kuten tietojen lisäämistä, poistamista ja muokkaamista varten luotiin omat PHP-funktiot. Seuraavassa tuote-sivun PHP-koodi, joka tutkii käyttäjän suorittamia toimintoja ja kutsuu tarvittavia funktioita (Koodi 20).

```

01 if ($_SERVER['REQUEST_METHOD']=='GET' || isset($_POST["tuotteet"]))
02 {
03     tulosta_tuotteet();
04     tuote_kentat();
05 }
06
07 if (isset($_POST["lisaa_tuote"])) {
08     lisaa_tuote();
09 }
10
11 if (isset($_POST["muuta"])){
12     muuta_tuote();
13     tulosta_tuotteet();
14     tuote_kentat();
15 }
16
17 if (isset($_POST["poista"])) {
18     poista_tuote();
19     tulosta_tuotteet();
20     tuote_kentat();
21 }
22
23 if (isset($_POST["muokkaa"])) {
24     tulosta_tuotteet();
25     muokkaa_tuote();
26 }

```

Koodi 20. Tuote-sivun PHP-koodi käyttäjän toimintojen suorittamiseen.

Käyttäjän klikatessa ”Tuotteet”-painiketta kutsutaan `tulosta_tuotteet()`-funktiota tulostamaan tuotteet sivulle. `Tuote_kentat()`-funktio tulostaa lomakekentät uuden tuotteen lisäämistä varten tuotetaulukon alapuolelle. Samoja funktioita kutsutaan myös käyttäjän avatessa sivu GET-metodilla, esimerkiksi syöttämällä tuote-sivun suora osoite osoiteriviin ja painamalla Enter. Tällä toiminnolla on haluttu estää mahdollisen tyhjän sivun tulostus käyttäjälle. Navigointi- sekä toimintopainikkeet välittävät painikkeen nimen POST-metodilla palvelimelle. Samaa ohjelmointilogiikkaa käytetään myös hallintasovelluksen muilla sivuilla eli asiakas-, tuoteryhmä-, sekä tilaussivulla. Esimerkiksi asiakas -sivulla tarkistetaan klikkasiko käyttäjä ”Asiakkaat”-painiketta, jolloin kutsutaan `tulosta_asiakkaat()`-funktiota tulostamaan asiakastiedot sivulle. Vastaavasti asiakas-sivulla on Koodi 20:n mukaisesti `asiakas_kentat()`-funktio, joka tulostaa lomakekentät uuden asiakkaan lisäämistä varten. Vastaavia yhteisiä funktioita joka sivulla ovat `lisaa()`, `muokkaa()` ja `poista()`-funktiot. Näitä funktioita sovelletaan sivukohtaisesti. Tuote-sivulla käytetään lisäksi `muuta_tuote()`-funktiota, jolla tuote voidaan poistaa myynnistä verkkokaupan sivuilta tai palauttaa se tarvittaessa takaisin myyntiin. Käytännössä hallintasovelluksen jokaisella sivulla on olemassa toimintopainikkeet tietojen lisäämiseen, poistamiseen sekä muokkaamiseen.

## 5.2 Sivuston rakenne

Hallintasivustosta pyrittiin tekemään helppokäyttöinen ja selkeä. Hallintasivuston runko toteutettiin verkkokauppasivuston kaltaisesti taulukoimalla ja ulkoasumäärittelyihin luotiin oma CSS-tyylitiedosto. Kiinteä navigointivalikko sijoitettiin sivuston oikeaan yläkulmaan, johon jäi sopivasti tyhjää tilaa sitä varten. Navigointivalikkoon luotiin linkit asiakas-, tuote-, tuoteryhmä- ja tilaus-sivuille sekä ylimmäiseksi oma painike uloskirjautumista varten.

Koska kaikki hallintajärjestelmän sivut sisältävät samoja JavaScripteja, CSS-tyylimääritelmän sekä tiettyjä paikallaan pysyviä web-elementtejä, käytettiin niiden sisällyttämiseen varsinaiselle sivulle PHP:n include()-funktiota. Sivulla käytettiin kolmea eri include-tiedostoa. Header.inc-tiedosto sisältää PHP-funktiot istunnon voimassaolon tarkistamisen eli varmistuksen siitä, että käyttäjä on kirjautunut sisälle sekä tarvittavat alkutagit <html> ja <head>. Header.inc luotiin omaksi include-tiedostoksi, koska istunnonkäsittelyyn tarvittavat PHP-funktiot täytyy suorittaa ennen varsinaista HTML-koodia. Tämän jälkeen HTML-dokumentteihin sisällytettiin connect.php -tiedosto, joka avaa yhteyden tietokantaan. Kolmas include-tiedosto valikko.inc sisältää kaikille sivuille yhtenäisen navigointivalikon, logon sekä tarvittavat JavaScriptit. Muut ulkoasuun liittyvät kuten taustakuvan, taulukoiden, painikkeiden, tekstien ja linkkien muotoilut toteutettiin CSS-tyylimääritelmän (tyyli.css) avulla. Seuraavassa ohjelmalistauksessa esimerkki hallintasivujen rakenteesta (Koodi 21).

```
01 <?php include("header.inc");?>
02 <title>Tuotteiden hallinta</title>
03 <link rel="stylesheet" type="text/css" href="tyyli.css" />
04 <?php
05 include("connect.php");
06 include("valikko.inc");
07 ...
08 Sivukohtaiset funktiot sekä muu sivukohtainen sisältö.
09 ...
10 ?>
11 </body>
12 </html>
```

Koodi 21. Sivulla toistuvat tiedot voidaan sisällyttää sivulle omina tiedostoinaan PHP:n include()-funktiolla.

### 5.3 Järjestelmään kirjautuminen

Väärinkäytön estämiseksi hallintajärjestelmään pääsee kirjautumaan vain sallitut käyttäjät. Järjestelmän ylläpitäjälle on luotu käyttäjätunnus ja salasana sisäänkirjautumista varten. Hallintasivustolla käyttäjätunnuksen ja salasanan tallentamiseen käytettiin PHP:n omaa istunnon hallintaa. Seuraavassa yksinkertaistettu esimerkki hallintajärjestelmän kirjautumissivun toteutuksesta.

```

01 <?php
02 session_start();
03 if (isset($_SESSION['user']) && isset($_SESSION['pw']))
04     {echo "Olet jo kirjautunut sisään";}
05
06 else {
07     if ($_SERVER["REQUEST_METHOD"]=="GET") {
08         session_unset();
09         echo "<form action='index.php' method='post'>";
10         echo "Käyttäjätunnus: ";
11         echo "<input type='text' name='user'><br/>";
12         echo "Salasana: ";
13         echo "<input type='password' name='pw'><br/>";
14         echo "<input type='submit' value='Kirjaudu sisään'>";
15         echo "</form>";
16     }
17
18     else {
19         if ($_POST['user']=="mattim" && $_POST['pw']=="abc123") {
20             $_SESSION['user']=$_POST['user'];
21             $_SESSION['pw']=$_POST['pw'];
22             header("Location: sisan.php");
23         }
24
25         else {
26             session_unset();
27             echo "Tunnus tai salasana väärin";
28         }
29     }
30 }
31 ?>

```

Koodi 22. Hallintasivuston käyttäjätunnuksen ja salasanan tallentaminen PHP:n istunnon hallintaan.

PHP:ssa istunto aloitetaan `session_start`-funktiolla. Edellisessä esimerkissä tarkistetaan ensin onko käyttäjätunnus ja salasana jo annettu eli onko istunto voimassa. Jos istunto on voimassa, tulostetaan käyttäjälle ilmoitus ”Olet jo kirjautunut sisään”. Muussa tapauksessa tulostetaan sisäänkirjautumislomake. Sisäänkirjautumislomakkeessa käyttäjältä kysytään tunnusta ja salasanaa, jotka luetaan `$_SESSION`-taulukkoon. Jos annetut tunnukset ovat oikein, ohjataan käyttäjä etusivulle PHP:n `header("Location: sisan.php")` -funktiolla. Muussa tapauksessa käyttäjälle tulostetaan virheilmoitus.

Jotta sisäänkirjautumista ei olisi mahdollista ohittaa, esimerkiksi syöttämällä selaimeen hallintasivuston etusivun suora osoite on istunnon voimassaolo varmistettava joka sivun alussa (Koodi 23). Jos istuntomuuttujat eivät ole voimassa, ohjataan käyttäjä sisäänkirjautumissivulle.

```
01 <?php
02 session_start();
03 if (!(isset($_SESSION['user'])) && !(isset($_SESSION['pw']))) {
04     session_unset();
05     header("Location: index.php");
06 }
07 ?>
```

Koodi 23. Istunnon voimassaolon tarkistus.

Uloskirjautuminen tapahtuu navigointivalikon ”Kirjautu ulos” -painikkeesta. Uloskirjautuminen varmistetaan käyttäjältä ponnahdusikkunassa ja hyväksymisen jälkeen käyttäjä ohjataan log\_out.php-sivulle. Log\_out.php-sivulla käyttäjätunnuksen ja salasanan sisältämät \$\_SESSION[’user’] ja \$\_SESSION[’pw’] istuntomuuttujat tyhjenetään unset()-funktioilla. Seuraavassa log\_out-sivun sisältö pääpiirteittäin (Koodi 24).

```
01 <?php
02 session_start();
03 unset($_SESSION['user']);
04 unset($_SESSION['pw']); ?>
05 <html>
06 <head>
07 <title>Olet kirjautunut ulos</title>
08 <meta http-equiv="refresh" content="2; url=index.php">
09 </head>
10 <body>
11 <h2>Olet kirjautunut onnistuneesti ulos järjestelmästä.</h2>
12 </body>
13 </html>
```

Koodi 24. Uloskirjautuminen suoritetaan log\_out-sivulla tyhjentämällä \$\_SESSION[’user’] ja \$\_SESSION[’pw’] istuntomuuttujat.

Rivillä 8 oleva <meta>-elementti ohjaa käyttäjän kahden sekunnin kuluttua uloskirjautumisesta takaisin sisäänkirjautumissivulle. Tätä ennen käyttäjälle tulostetaan teksti onnistuneesta uloskirjautumisesta.

## 5.4 Tuote-sivu

Tuote-sivulla tuotteiden tiedot haetaan tietokannasta valinnaisten ehtojen mukaisesti ja tulostetaan taulukkoon. Tuotetaulukon yläpuolelle luotiin alasetolistasta, josta käyttäjä voi valita minkä tuoteryhmän tuotteet sivulle tulostetaan. Tämä helpottaa tuotteiden katselua etenkin jos tuotteita on paljon. Tuoteryhmien nimet haetaan alasetolistasta tietokannan tuoteryhmä-taulusta MySQL-komennolla. Jos käyttäjä lisää uuden tuoteryhmän tietokantaan tuoteryhmä-sivulta tulee se myös näkyviin tuote-sivun alasetolistaan. Tuoteryhmän nimi välitetään alasetolistasta GET-metodilla ja yhdistetään MySQL-kyselylauseeseen. Seuraavassa koodilistauksessa on esitelty tuotetietojen haku tietokantakyselyllä sekä tuotteiden valitseminen alasetolistan ryhmänimen mukaan.

```
01 $query = mysql_query("SELECT tuoteid, ryhmanimi, tuotenimi,
02 tuoteseloste, poistettu, hinta, 0,78*hintaa AS alvhinta FROM tuote,
03 tuoteryhma WHERE tuote.ryhmaid = tuoteryhma.ryhmaid AND
04 tuoteryhma.ryhmanimi = '$_GET[ryhma]' ORDER BY tuoteid");
```

Koodi 25. Tuotetietojen haku tuote-sivulle alasetolistasta valitun ryhmänimen mukaan.

Tuotetaulukon haetaan tietokannasta tuotteen id-numero, tuoteryhmän nimi, tuotteen nimi, tuoteseloste, poistettu-kenttä ja hinta. Kaikki muut tiedot voidaan kerätä tuotetaulusta paitsi tuoteryhmän nimi, joka haetaan tuoteryhma-taulusta ja arvonlisävero, joka lasketaan tuotteen hinnan perusteella. Rivillä 4 oleva \$\_GET[ryhma] on alasetolistasta PHP-tulkille välitetyn tuoteryhmän nimi. Jos tietty ryhmänimi on valittu alasetolistasta, tulostetaan tuote-sivulle tietyn tuoteryhmän tuotteet määrittämällä tuoteryhma.ryhmanimi = \$\_GET[ryhma]. Tuotteet järjestetään automaattisesti tuotteen id-numeron mukaan.

Taulukon tiedot voidaan sarakekohtaisesti järjestää JavaScriptin avulla nousevaan tai laskevaan järjestykseen esimerkiksi tuotteiden hinnan mukaan klikkaamalla hintasarakkeen otsikkoa. Jokaisen tuoterivin perään tulostetaan lisäksi Muuta, Poista-, Muokkaa- ja Avaa-painikkeet, jotka toimivat POST-metodilla. Muuta-painikkeella tuotteen poistettu-kentän arvo voidaan muuttaa 0:ksi tai 1:ksi, joka määrittää tuotteen näkyvyyden verkkokaupassa. Poista-painikkeella voidaan poistaa ko. rivillä oleva tuote tietokannasta, mikäli se ei riko viite-ehyettä. Muokkaa-painikkeella käyttäjä voi muuttaa ko. rivillä olevan tuotteen tietoja. Tiedot tulostetaan muokattaviksi sivun alareunassa oleviin lomakekenttiin. Avaa-painikkeella käyttäjä voi avata tuotteen esittelysivun muokkauk-



seen. Kaikki hallintasivuston kautta tuotteisiin tehdyt muutokset tulevat voimaan myös verkkokaupan sivuille, koska muutokset tehdään samaan tietokantaan. Seuraavassa kuvassa näkymä hallintasovelluksen tuote-sivusta.

**Fin 158**

Kirjautu ulos

Asiakkaat  
Tuotteet  
Tuoteryhmät  
Tilaukset

TUOTTEET

Valitse tuoteryhmä: ▼

**Vesiautomaatit**

#	Tuoteryhmä	Tuotenimi	Tuoteseloste	Hinta	ALV	Myyntissä	Muokkaa	Poista	Sivu	
4	Vesiautomaatit	Aqua Filter FDL	Vesijohtoverkkoon liitettävä lattiamallinen vesiautomaatti. Helppokäyttöinen ja samalla tyylikäs raikasvesiautomaatti, joka lisää viihtyisyyttä missä vaan. Vesiautomaatista saa raikasta, jäähdytettyä (4 – 12 C°) sekä kuumaa vettä (85 – 100 C°), josta saa vaivattomasti lämpimiä juomia. Lisäksi vielä huoneenlämpöistä vettä (n. 21 C°). Käyttökohteina taukottilat, neuvottelutilat ja toimistot. Aqua Filter on vesijohtoverkkoon liitettävä vesiautomaatti.	1342.00	295.24	Kyllä	Muuta	Muokkaa	Poista	Avaa
5	Vesiautomaatit	Aqua Filter FDS	Vesijohtoverkkoon liitettävä pöytämallinen vesiautomaatti. Helppokäyttöinen ja samalla tyylikäs raikasvesiautomaatti. Toimii vesianostelijana jäähdytetyille, kuumennetulle ja huoneenlämpöiselle vedelle. Lisävarusteena saatavana erilaisia tiskin alle sijoitettavia suodatinvaihtoehtoja. Erinomainen ratkaisu ruokaloiden palvelinjoille. Vesijohtoverkkoon liitettävä vesiautomaatti.	939.40	206.67	Kyllä	Muuta	Muokkaa	Poista	Avaa
19	Vesiautomaatit	Connect UV 23 A	Vesijohtoverkkoon liitettävä lattiamallinen vesiautomaatti Huoltohälytysjärjestelmällä ja annostelualueen UV valolla Laite soveltuu kohteisiin, joissa vaaditaan erityistä puhtautta.	1464.00	322.08	Kyllä	Muuta	Muokkaa	Poista	Avaa
20	Vesiautomaatit	Rio 23 A-C	Vesijohtoverkkoon liitettävä lattiamallinen vesiautomaatti paikkaan missä ihmiset kaipaavat virkistystä, jossa vedenkulutus on suurta.	1220.00	268.40	Kyllä	Muuta	Muokkaa	Poista	Avaa
21	Vesiautomaatit	Rio Jetset	Vesijohtoverkkoon liitettävä pöytämallinen vesiautomaatti. Tiskin päällä näytävä vesiautomaatti kahviloihin ja ruokaloihin.	671.00	147.62	Kyllä	Muuta	Muokkaa	Poista	Avaa

Tuotteen nimi:

Tuoteseloste:

Hinta:

Tuoteryhmä: Ilmanvientäjät ▼

Lisää tuote  
Tyhjennä

Kuva 12. Näkymä verkkokaupan hallintasovelluksen tuote-sivulta.

Hallintasovelluksen asiakas-, tuoteryhmä- ja tilaus-sivuilla tietojen haku ja tulostus taulukkoon on lähes samankaltainen kuin tuote-sivulla. Eroavaisuus tietojen haussa sivujen välillä keskittyy lähinnä tietojen hakuun eri tauluista. Esimerkiksi asiakas-sivulla asiakkaan tiedot haetaan luonnollisesti asiakas-taulusta.

## 5.5 Tuotteen esittelysivun luonti

Verkkokaupassa jokaisella tuotteella on oma esittelysivunsa, joka voi sisältää kuvia tuotteesta, tarkemman tuoteselostuksen tai vaikka linkin tuote-esitteeseen. Käsin tällaisten sivujen tekeminen usealle tuotteelle on työlästä, koska jokaiselle tuotteelle pitäisi tehdä oma esittelysivunsa ja tallentaa ne web-palvelimelle. Tämän vuoksi hallintajärjestelmään tehtiin oma tuotesivugeneraattori helpottamaan sivujen luontia ja päivittämistä.

Tuotteen esittelysivu voidaan avata muokattavaksi klikkaamalla tuote-sivulla tietyn tuoterivin perästä ”Avaa”-painiketta. Tuotteen esittelysivu avataan muokattavaksi lisää\_tiedot.php-sivulla. Tuotteen tiedot haetaan lisää\_tiedot-sivulle tietokannan tuotetaulusta tuotteen id-numeron perusteella. Tuotteen id välitetään lisää\_tiedot-sivulle GET-metodilla seuraavan esimerkin mukaisesti.

```
01 <input type="button" value="Avaa" onClick=
02 "MM_openBrWindow('lisaa_tiedot.php?id=<?echo $id?>', 'pop')" />
```

Koodi 26. Esittelysivu avataan tuotteen id-numeron perusteella, joka välitetään GET-metodilla.

Lisää\_tiedot.php-sivulla tuotteelle voidaan lisätä tarkka tuoteseloste sekä kuva. Tehdyt muutokset nähdään sivulla heti kun tiedot on lisätty. Seuraavassa kuva erään tuotteen muokkaussivusta (Kuva 13).



**Rio Jetset**

Vesijohtoverkkoon liitettävä pöytämallinen vesiautomaatti  
Tiskin päällä näyttävä vesiautomaatti kahviloihin ja ruokaloihin. Laitteista saa  
jähdytettyä ja huoneen lämpöistä vettä suoraan vesijohtoverkosta.  
Kun laite varustetaan tiskin alle sijoitettavilla suodattimilla, jää tiskin päälle  
ainoastaan veden annostelija.

- \* Jähdytyskapasiteetti 22 l / h.
- \* Mitat K/L/S: 470/490/390mm
- \* Sähköteho: 130 W

Kuva 13. Muokkaussivulta tuotteelle voidaan lisätä tarkka tuoteseloste ja kuva.

Haluttu kuva valitaan tietokoneelta ”Selaa”-painikkeella ja tallennetaan palvelimelle käyttäen PHP:n tiedostonkäsittelyfunktioita. Jotta kuvan lähetys onnistuu, täytyy myös lomakkeen ominaisuudet olla määriteltynä oikein (Koodi 27).

```
01 <form enctype="multipart/form-data" action="" method="POST">
02   <input type="hidden" name="MAX_FILE_SIZE" value="200000" />
03   <input name="pic" type="file" />
04   <input type="submit" value="Lisää kuva" name="kuva" />
05 </form>
```

Koodi 27. Lomakkeen enctype-attribuutti on määriteltävä oikein, jotta kuvan lähetys onnistuu.

Lomakkeen enctype-attribuutti tulee olla "multipart/form-data" tai muuten kuvan lähetys ei toimi. Myös tiedoston hyväksyttävä maksimikoko tavuina voidaan määrittellä kentän ominaisuudeksi, jolloin PHP-tulkki tarkastaa asetetun kokorajoituksen. Kuvan siirto palvelimelle ei onnistu, jos käyttäjän valitseman tiedoston koko ylittää rajoituksen. Esimerkissä kuvan maksimikooksi on määritelty 200 kilotavua. Tiedostoja käsiteltäessä lomakekentän tyyppi on määriteltävä ”file”. Kentän nimeksi on annettu ”pic”, johon viitataan myöhemmin PHP:n tiedostonkäsittelyfunktioissa.

Seuraavassa koodissa on kuvan lisäämiseen ja nimeämiseen tarvittavat funktiot.

```
01 $uploaddir = '/var/www/fin158/tuotteet/';
02 $uploadfile = $uploaddir . basename($_FILES['pic']['name']);
03
04 if(isset($_POST['kuva'])) {
05   if (move_uploaded_file($_FILES['pic']['tmp_name'], $uploadfile)){
06     rename("../tuotteet/".$_FILES['pic']['name']."",
07     "../tuotteet/".$tuoteid.".jpg");
08   }
09   else {
10     echo "<p>Kuvan lisäys ei onnistunut!</p>\n";
11   }
12 }
```

Koodi 28. Kuvan tallentaminen ja nimeäminen web-palvelimelle.

Koodi 28:n alussa \$uploaddir-muuttujassa on määritelty kuvan tallennushakemisto ja \$uploadfile-muuttujassa alkuperäisen kuvan nimi. Rivin 4 ehtolause tarkastaa onko kuva valittu lähettämistä varten, jolloin jatketaan kuvan tallentamista palvelimelle. Jos kuva yritetään tallentaa, vaikkei sitä ole valittu, tulostetaan käyttäjälle virheilmoitus (Rivit 9-11). Rivin 5 \$\_FILES['pic']['tmp\_name'] -muuttuja sisältää palvelimelle kirjoitetun väliaikaistiedoston hakemistopolkuineen. Jos kuvan siirto onnistuu, siirretään

kuva verkkokauppasivuston alla olevaan "tuotteet"-nimiseen hakemistoon. Kuva nimeetään rename()-funktiolla tuotteen id-numeron mukaisesti .jpg -päätteiseksi. Esimerkiksi tuotteen, jonka id on 3, kuva tallennetaan palvelimelle nimellä 3.jpg (Rivit 6-7). Näin tuotetta vastaava kuva on helppo hakea esittelysivulle, koska tuotteen sekä kuvan id-numero on sama.

Tuotteen tarkempi kuvaus voidaan kirjoittaa lomakkeen textarea-kenttään ja tallentaa tuote-tauluun ”Lisää seloste” -painikkeesta. Seuraavassa koodilistauksessa on esitelty kuvauksen päivittäminen tuote-tauluun.

```
01 if(isset($_POST['kuvaus'])) {
02     $kuvaus = stripslashes($_POST['kuvaus']);
03     $query = mysql_query("UPDATE tuote SET kuvaus = '$kuvaus' WHERE
04     tuoteid = ".$tuoteid."")
05 }
```

Koodi 29. Kuvauksen päivittäminen tuote-tauluun.

Koodin alussa tarkistetaan onko kuvaus-lomaketta lähetetty. Tämän jälkeen käyttäjän antama tuotekuvaus luetaan \$kuvaus-muuttujaan ja samalla stripslashes()-funktiolla poistetaan mahdolliset kenoviivat heitto- ja lainausmerkkien edestä (Rivi 2). Kuvaus päivitetään tuote-tauluun MySQL:n UPDATE-komennolla oikean tuotteen kohdalle tuotteen id-numeron perusteella.

## 5.6 Tietojen lisäys ja muokkaus

Tietojen lisäys- ja muokkaustoiminnot ovat välttämättömiä etenkin tuotteiden ja tuoteryhmiä hallinnassa. Näitä toimintoja tarvitaan, kun halutaan lisätä uusia tai muokata jo olemassa olevia tuoteryhmiä ja niihin kuuluvia tuotteita. Lisäys- ja muokkaustoiminnot ovat myös käytettävissä asiakas- ja tilaussivulla. Tavallisesti uusia asiakas- ja tilaustietoja ei tarvitse lisätä, koska tilausvaiheessa nämä tiedot kirjataan automaattisesti asiakas-, tilaus- ja tilausrivitauluihin. Kuitenkin erityistapauksissa muutoksia tai lisäyksiä voi joutua tekemään käsin asiakas- ja tilaustietoihin esimerkiksi asiakkaan halutessa muuttaa jo tekemäänsä tilaustaan. Tietojen lisäys ja muokkaus toimii käytännössä samalla periaatteella niin asiakas-, tuote-, tuoteryhmä- kuin tilaussivullakin. Ainoa ero on kohdistaa tietojen lisäykset ja muutokset tietokannan oikeaan tauluun sivukohtaisesti. Jokaisella sivulla tulostetaan omat sivukohtaiset lomakekentät tietojen lisäämistä ja muokkaamista varten. Seuraavissa kappaleissa on esitelty tuotteiden lisäys ja muokkaus.

Uusia tuotteita tietokantaan voi lisätä tuote-sivun alareunassa olevalla lomakkeella (Kuva 14). Pakolliset tiedot ovat tuotteen nimi, tuoteseloste, tuotteen hinta sekä tuoteryhmä, johon tuote kuuluu. Tuoteryhmät haetaan tietokannan tuoteryhma-aulusta ja tulostetaan lomakkeen alavetolistaan. Seuraavassa kuva tuotteen lisäyslomakkeesta.

Tuotteen nimi:	<input type="text" value="Connect UV 23 A"/>
Tuoteseloste:	<input type="text" value="Vesijohtoverkkoon liitettävä lattiamallinen vesiautomaatti Huoltohälytysjärjestelmällä ja annostelualueen UV valolla Laite soveltuu kohteisiin, joissa vaaditaan erityistä puhtautta."/>
Hinta:	<input type="text" value="1299.00"/>
Tuoteryhmä:	<input type="text" value="Vesiautomaatit"/>
<input type="button" value="Lisää tuote"/>	
<input type="button" value="Tyhjennä"/>	

Kuva 14. Uuden tuotteen lisäyslomake hallintasovelluksen tuote-sivulla.

Uusi tuote saadaan lisättyä tietokannan tuote-tauluun lomakkeen ”Lisää tuote” -painikkeella. Tyhjennä-painikkeella lomakkeen kentät voidaan tarvittaessa tyhjentää. ”Lisää tuote” -painike kutsuu lisää\_tuote()-funktiota, joka tarkistaa ensin käyttäjän antamat syötteet virheiden varalta ja suorittaa sitten tarvittavat MySQL-komennot uuden tuotteen lisäämiseksi tietokannan tuote-tauluun. Lomakkeen kaikki kentät tulee olla täytettynä ja lisäksi tuotteen hinnaksi voidaan hyväksyä vain numeerisia lukuja sekä desimaalipiste. Tarkastamiseen käytettiin PHP:n eregi()-funktiota, kuten tilaustietojen oikeellisuuden varmistamisessa kappaleessa 4.8. Virheelliset kentät ilmoitetaan käyttäjälle punaisella kentän nimellä ja hyväksytyt tiedot säilytetään lomakkeen kentissä PHP:n muuttujien avulla. Seuraavassa esimerkissä lisää\_tuote()-funktion suorittama tuotteen nimen tarkistus.

```

01 function lisaa_tuote() {
02     $errors = NULL;
03
04     if(isset($_POST['nimi']) && strlen($_POST['nimi']) &&
05     eregi('^[a-ö 0-9//-/]+$' , ($_POST['nimi']))) {
06         $nimi = mysql_real_escape_string(stripslashes($_POST['nimi']));
07         $ots_nimi = 'Tuotteen nimi: ';
08     }
09     else {
10         $errors .= 'Kenttä puuttuu';
11         $ots_nimi .= '<font color="red">Tuotteen nimi</font>: ';
12     }
13 }
...

```

Koodi 30. Tuotelomakkeen tuotenimen oikeellisuuden tarkistus PHP:n eregi()-funktiolla.

Kun tietojen oikeellisuus on varmistettu, suoritetaan tarvittavat MySQL-komennot tuotteen lisäämistä varten (Koodi 31).

```

01 $query = mysql_query("insert into tuote values ('$tuoteid',
02 '$ryhmaid', '$nimi', '$seloste', '$hinta')");
03 or die ("Toiminto ei onnistunut - '$query': ".mysql_error());

```

Koodi 31. MySQL-komento tuotteen lisäämiseksi tietokannan tuote-tauluun.

Muokkaa-painiketta painettaessa kutsutaan tuote\_kentat\_m()-funktiota, joka lukee tuotteen nykyiset tiedot kuten tuotteen nimen, hinnan sekä tuoteselosteen PHP:n muuttujiin ja tulostaa ne valmiiksi muokattaviksi lomakekenttiin (Koodi 32). Näin käyttäjän ei tarvitse pieniä muokkauksia varten kirjoittaa kaikkia tuotetietoja uudelleen lomakkeeseen.

Koodi 32:ssä on ote muokkaa\_tuote()-funktion sisällöstä.

```

01 $query = mysql_query("SELECT hinta FROM tuote WHERE id =
02 ".$_POST['id']."");
03 echo '<tr><td><p>Hinta: </p></td><td><input type="text"
04 name="hinta" value="'.mysql_result($query, "id").'"></td></tr>';

```

Koodi 32. Tuotetietojen haku lomakekenttiin muokkausta varten.

Tuotteen hinta haetaan tietokannasta tuotteen id:n perusteella, joka välitetään POST-metodilla muokkaa-painiketta painettaessa. Haku suoritetaan tietokannasta PHP:n mysql\_query()-funktiolla ja tulos tulostetaan mysql\_result()-funktiolla lomakkeen hintakentän arvoksi.

## 5.7 Tietojen poisto

Tietojen poistamiseksi tietokannasta luotiin oma funktio, jota sovelletaan asiakas-, tuote-, tuoteryhmä- ja tilaussivulla. Poisto-toimintoa ei tulisi käyttää kuin erityistapauksissa esimerkiksi poistettaessa vahingossa lisättyjä tuotteita tai peruttuja tilauksia. Poisto-toiminto poistaa tiedot kokonaan tietokannasta, jolloin viite-eheyden puuttuessa esimerkiksi tilauksiin voisi jäädä viittauksia olemattomiin tuotteisiin. Poisto-toiminto on toteutettu samalla tavalla joka sivulla. Käytännössä poistaminen suoritetaan yhdellä MySQL-lauseella, joka kohdistaa poistamisen tiettyyn tietokannan tauluun (Koodi 33). Seuraavassa koodilistauksessa on esimerkki tuotteiden poistamisen toteutuksesta poista\_tuote()-funktion avulla.

```
01 function poista_tuote() {
02     mysql_query("DELETE FROM tuote WHERE tuoteid=" . $_POST['id'] . " ");
03 }
```

Koodi 33. Poista\_tuote()-funktio sisältää MySQL-komennon tuotteen poistamiseksi tietokannasta.

Poista\_tuote()-funktion rivillä 2 suorittama MySQL-komento poistaa tuotteen tietokannan tuote-taulusta. Samalla tuote poistuisi myös näkyvistä verkkokaupan sivuilta, mikäli tuote olisi myynnissä. Tämän jälkeen tuotteet ja tuotekentät tulostetaan päivitettyinä sivulle. Oikea poistettava tuote tunnistetaan id-numeron perusteella, joka välitetään POST-metodilla käyttäjän klikatessa poista-painiketta tietyn tuotteen kohdalta. Poistettaessa tietoja esimerkiksi asiakas-taulusta vaihdetaan mysql\_query()-funktion sisällöksi "DELETE FROM asiakas WHERE asiakasid=" . \$\_POST['id'] . " "). Tilaus-sivulla poisto-toiminto on asteen monimutkaisempi, koska poistaminen voidaan kohdistaa tiettyyn tilausriviin eli tilattuun tuotteeseen tai koko tilaukseen. Koko tilausta poistettaessa tiedot täytyy poistaa sekä tilaus-taulusta että tilausta vastaavasta tilausrivi-taulusta.

Tuotteiden poistamiseksi verkkokaupasta tuote-tauluun luotiin poistettu-kenttä, jonka arvoksi voidaan hallintasivulla määrittää 1, jolloin tuote poistetaan näkyvistä vain verkkokaupasta. Oletusarvoisesti uuden tuotteen poistettu-kentän arvo on 0, jolloin tuote tulostetaan näkyviin verkkokauppasivustolle.

## 6 YHTEENVETO

Verkkokaupan toteutus oli pidempi prosessi, kuin aluksi olisin voinut kuvitella. Eniten aikaa kului huolellisen ja toimivan koodin tuottamiseen sekä testaamiseen. Verkkokauppa otettiin käyttöön riittävän testaamiseen jälkeen, ja sen kautta on jo tehty muutamia tilauksia. Testaamiseen käytettiin runsaasti aikaa, jottei varsinaisessa käytössä esiintyisi ongelmia. Verkkokaupasta tuli toimiva ja suunnitellun mukainen, vaikka projektilla olikin matkan varrella taipumusta paisua liian laajaksi. Samalla mieleen tuli kuitenkin vartenotettavia ajatuksia verkkokauppasovelluksen jatkokehittämiseen.

Jatkossa tarkoituksena on parantaa tuotteiden näkyvyyttä verkkokaupassa järkevemmän esillepanon kautta. Tuotteiden esittelysivut voisi toteuttaa ilman ponnahdusikkunoiden käyttöä ja muuttaa nykyisen taulukkomaisen tuotelistauksen visuaalisemmaksi. Tietokantaan suunniteltua asiakas-taulua voi hyödyntää tulevaisuudessa verkkokaupan rekisteröitymisominaisuutta varten. Asiakkaat voisivat halutessaan rekisteröityä verkkokaupan käyttäjiksi, jolloin heillä olisi mahdollisuus muun muassa seurata tilaushistoriaansa sekä muuttaa omia tilaustietojaan.

Verkkokaupan hallintajärjestelmästä tuli oiva apu verkkokauppasivuston ylläpitämiseen. Tuotesivujen muokkausta on kuitenkin tarkoitus vielä tehdä helppokäyttöisemmäksi ja laajentaa hallintajärjestelmän ominaisuuksia.

Kaiken kaikkiaan verkkokaupaprojekti yllätti laajuudellaan, mutta soveltui mielestäni hyvin opinnäytetyön aiheeksi monipuolisuutensa ansiosta. Projektin varrella opin paljon uutta PHP- ja MySQL-ohjelmoinnista sekä toimivan web-liittymän toteutuksesta. Verkkokauppaa varten perustetun Linux web-palvelimen pystyttämisen ja ylläpidon ohessa tutuksi tulivat myös Apache HTTP -palvelinsovelluksen konfigurointi, Awstats-tilastointisovellus sekä Squirrelmail-sähköpostisovellus.



## LÄHDELUETTELO

### **Julkaisut:**

Korpela, J. 2003. CSS-tyylit. Porvoo: Docendo Finland Oy.

Peltomäki, J. & Nykänen, O. 2006. Web-selain-ohjelmointi. Porvoo: Docendo Finland Oy.

Rantala, A. 2005. Web-ohjelmointi. Porvoo: Docendo Finland Oy.

### **Verkkodokumentit:**

Ekonoja, A., Lahtonen, T. & Mäntylä J. 2007. Lomakkeiden käsittelytekniikat. [verkkodokumentti] Jyväskylä: Jyväskylän yliopisto [Viitattu 7.11.2007]  
Saataavissa: <http://appro.mit.jyu.fi/www/luennot/luento8>

Lahtonen, T. 2004. Relaatiotietokannat. [verkkodokumentti] Jyväskylä: Jyväskylän yliopisto [Viitattu 7.11.2007]  
Saataavissa: <http://appro.mit.jyu.fi/doc/tiedonhallinta/tietokannat>

Langridge, S. 2007. sortable: Make all your tables sortable [verkkodokumentti]. [Viitattu 12.4.2007]. Saataavissa: <http://www.kryogenix.org/code/browser/sortable/>

Korpela, J. 2006. Dynaaminen HTML. [verkkodokumentti] Tampere: Tampereen teknillinen yliopisto [Viitattu 25.1.2006].  
Saataavissa: <http://www.cs.tut.fi/~jkorpela/webjulk/3.6.html>

Ohjelmointiputka 2007: Käytännön PHP-opas [verkkodokumentti]. [Viitattu 15.6.2007]. Saataavissa: <http://www.ohjelmointiputka.net/opas.php?tunnus=phpj4>

Sundström, L. 1998. Laitteet, verkkoyhteys. [verkkodokumentti] Espoo: Teknillinen korkeakoulu [Viitattu 12.6.2007].  
Saataavissa: <http://www.netlab.tkk.fi/opetus/s38118/s98/htyo/51/Laitteet.shtml>

Tuikka, T. 2006. Internet-ohjelmointi. [verkkodokumentti] Jyväskylä: Jyväskylän ammattikorkeakoulu [Viitattu 28.1.2006].  
Saataavissa: [http://sinuhe.jypoly.fi/~tuito/i\\_opinnot/main3.htm](http://sinuhe.jypoly.fi/~tuito/i_opinnot/main3.htm)

Wikipedia 2007: Apache [verkkodokumentti]. [Viitattu 12.6.2007]. Saataavissa: <http://fi.wikipedia.org/wiki/Apache>

Wikipedia 2006: JavaScript [verkkodokumentti]. [Viitattu 28.1.2006]. Saataavissa: <http://fi.wikipedia.org/wiki/JavaScript>

Wikipedia 2006: MySQL [verkkodokumentti]. [Viitattu 11.2.2006]. Saataavissa: <http://fi.wikipedia.org/wiki/MySQL>

Wikipedia 2006: PHP [verkkodokumentti]. [Viitattu 28.1.2006]. Saataavissa: <http://fi.wikipedia.org/wiki/PHP>