

Saimaan ammattikorkeakoulu
Tekniikka Imatra
Sähkötekniikka

Juhana Timperi

VARAOSAKIRJANPITO WEB-SOVELLUKSENA

Opinnäytetyö 2010

TIIVISTELMÄ

Juhana Timperi

Varaosakirjanpito web-sovelluksena, 24 sivua, 3 liitettä

Saimaan ammattikorkeakoulu, Imatra

Tekniikka, Sähkötekniikka

Ohjaaja: DI Jouni Könönen

Opinnäytetyön tarkoituksena oli toteuttaa varaosakirjanpitosovellus lappeenrantalaiselle sähköryitykselle. Opinnäytetyössä tutustuttiin eri ohjelmointikieliin tehtäessä web-sovelluksia.

Tutkimusmenetelminä käytettiin internetistä löytyviä ohjelmointioppaita ja valmiita malliesimerkkejä. Työn teoriaosuudessa keskityttiin kuvaamaan työn eri vaiheita ja käytettyjä ohjelmia ja ohjelmointikieliä.

Opinnäytetyö toteutettiin keväällä 2010. Työ aloitettiin suunnittelulla ja aihepiirin rajauksella. Tämän jälkeen aloitettiin varsinainen ohjelmointi ja sovelluksen testaus. Kun sovellus saatiin toimimaan, otettiin se käyttöön Aatoskone Oy:ssä.

Tuloksien perusteella voidaan todeta, että yksinkertaisen web-sovelluksen teko onnistuu keneltä tahansa, joka osaa jotakin ohjelmointikieltä. Opinnäytetyöllä luotiin perusjärjestelmä, jota tullaan jatkossa kehittämään eteenpäin.

Avainsanat: Varaosakirjanpito, tietokanta, ohjelmointi

ABSTRACT

Juhana Timperi

Sparepart accounting by webprogramming. 24 pages, 3 appendices

Saimaa University of Applied Sciences

Electrical engineering

Bachelor's thesis 2010

Instructor: M.Sc. Jouni Könönen

The aim of this thesis was to made sparepart accounting to electric company.

The study methods used were an internet sources and programming guide books.

The thesis was made during the spring 2010.

Key words: Accounting, database, programming

SISÄLTÖ

1 JOHDANTO.....	5
2 WEB-OHJELMOINTI.....	5
PHP-ohjelmointi.....	6
3 RELAATIOTIETOKANNAT.....	8
3.1 Yleistä tietoa tietokannoista.....	9
3.2 Käytännön sovelluksia tietokannoista.....	9
3.3 MySQL.....	9
3.4 Taulujen perustaminen MySQL:ään.....	11
4 SIVUJEN TOTEUTUS.....	13
4.1 Työskentely MySQL-PHP -yhdistelmällä.....	13
4.2 Käytetyt ohjelmat.....	13
4.3 Työn kulku.....	14
4.4 Varaosakirjanpidon ominaisuudet.....	16
4.5 Suojattu hallintasivu.....	18
4.6 Sivuston ulkoasu.....	20
5 YHTEENVETO.....	22
LÄHTEET.....	23

LIITTEET

Liite 1 Koodi, joka näyttää tilatut osat

Liite 2 Koodi, joka tekee lisäyksen tietokantaan

Liite 3 Kuva sovelluksesta

LUETTELO KÄSITTEISTÄ

- HTML = Hypertext Markup Language, kieli, josta internetsivut rakentuvat.
- PHP = Hypertext Preprocessor, ohjelmointikieli, jota käytetään erityisesti web-ympäristössä.
- SQL = Structured Query Language, kyselykieli, jolla voidaan tehdä hakuja relaatiotietokannoista.
- APACHE = Avoimeen lähdekoodin perustuva palvelinohjelma.
- CSS = Cascading Style Sheets, tyyliohjejärjestelmä, jolla dokumentille voi määritellä useita tyyliohjeita.
- JavaScript = Pääasiassa web-ympäristössä käytettävä komentosarjakieli.
- Perl = Practical Extraction and Report Language, skriptimäinen ohjelmointikieli (Wikipedia, 6.)

1 JOHDANTO

Opinnäytetyöni käsittelee web-ohjelmointia ja yksinkertaista tietokantasovellusta. Toteutan opinnäytetyönäni varaosakirjanpitosovelluksen, joka tulee lappeenrantalaisen sähköyrityksen, Aatoskone Oy:n, käyttöön. Aatoskone Oy on Kaakkois-Suomen alueella toimiva sähkölaitehuoltoihin erikoistunut yritys. Olen toiminut yrityksen palveluksessa vuodesta 2005 lähtien. Tammikuussa 2010 minut palkattiin yritykseen huoltopäälliköksi ja samalla minusta tuli yrityksen osakas.

Toteutan opinnäytetyöni siten, että Aatoskone Oy saa varaosajärjestelmänsä selkeäksi ja käyttäjäystävälliseksi. Tähän asti varaosat on kirjattu tavalliselle paperilehtiölle, joten opinnäytetyöni päivittää samalla yrityksen toimintatapoja 2000-luvulle.

Tarkoitukseni on saada pieni ja kevyt vaihtoehto maksullisille varaosasovelluksille. Käsittelem opinnäytetyössäni vain PHP:tä ja MySQL:ää, koska PHP on jo itsessään niin laaja kokonaisuus, että esimerkiksi JavaScriptin mukaanotto tekisi aiheesta aivan liian laajan. Keskityn opinnäytetyössäni rakentamaan toimivan ja selkeän varaosatiekantasovelluksen mahdollisimman pienillä kustannuksilla. Aatoskone Oy toimii usean eri laitevalmistajan virallisena huoltoliikkeenä, joten tilattavien ja toimitettavien varaosien määrä on erittäin suuri. Yrityksellä ei ole varaosavastaava, joten jokainen työntekijä tekee ja lähettää varaosatilauksia. Tästä johtuen sovelluksen selkeys ja toimivuus ovat tärkeitä asioita.

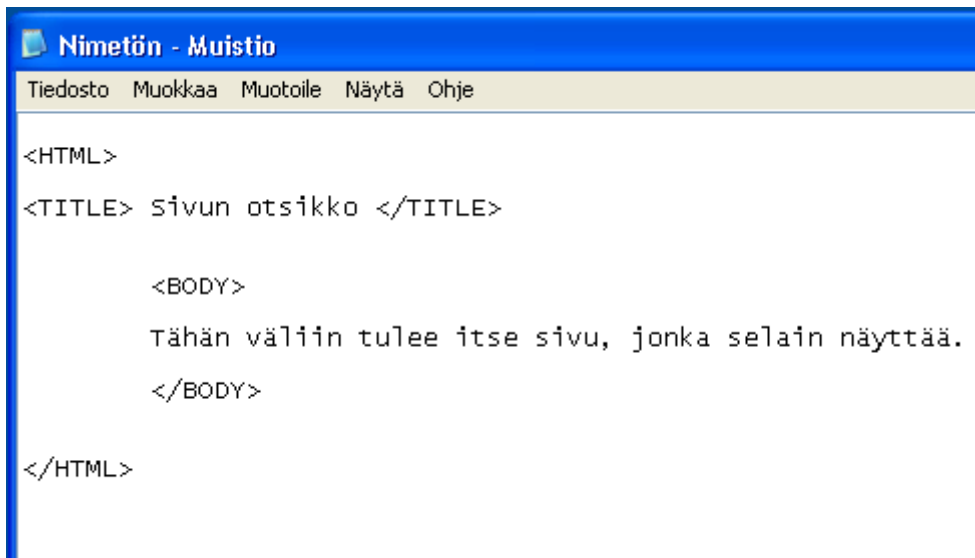
2 WEB-OHJELMOINTI

Web-ohjelmoinnin perustana on se, että käytännössä kaikki Internetissä tapahtuvat asiat vaativat toimiakseen yhden tietyn kielen, HTML:n.

HTML on WWW-sivujen kuvauskieli, jolla kuvataan sivujen ja tekstin rakennetta. Kotisivujen tai muiden web-sovellusten tekoa aloitteleva käyttäjä törmää siis

ensimmäisenä HTML-kieleen. Perussivut, joilla on pelkästään tekstiä, kuvia ja linkkejä, voidaan tehdä HTML:llä.

Seuraavassa kuvassa esitetään yksinkertaiselta näyttävä, mutta täysin toimivan internetsivun lähdekoodi. Tämän monimutkaisempaa ei tarvita:

A screenshot of a Notepad window titled "Nimetön - Muistio". The window has a menu bar with "Tiedosto", "Muokkaa", "Muotoile", "Näytä", and "Ohje". The main text area contains the following HTML code:

```
<HTML>
<TITLE> sivun otsikko </TITLE>

    <BODY>
    Tähän väliin tulee itse sivu, jonka selain näyttää.
    </BODY>

</HTML>
```

Kuva 1. WWW-sivun rakenne.

Vuorovaikutteisten sovellusten, kuten palautelomakkeiden tekoon HTML ei sovi, sillä sen tehtävänä on pelkästään esittää tekstin tyylikorostukset, kuten lihavoinnit, rivinvaihdot ja otsikot. Ohjelmointikielille tyypillisiä funktioita tai muuttujia HTML:ssä ei ole, vaan perusideana on kirjoittaa teksti aina aloitusmerkin ja lopetusmerkin väliin. HTML:ää voidaan kirjoittaa millä tekstieditorilla tahansa ja notepad (muistio) onkin aivan riittävä ohjelma.

PHP-ohjelmointi

PHP on palvelinpuolella suoritettava komentosarjakieli. Se muistuttaa hyvin paljon Perliä, C:tä ja Javaa, mutta sillä on myös paljon aivan omia, uniikkeja, piirteitä. PHP soveltuu ominaisuuksiensa takia mainiosti HTML:n sekaan. PHP toimii niin sanotusti upotetussa tilassa, eli sitä voidaan kirjoittaa suoraan kaikkien HTML-sivujen lähdekoodiin. (KK Mediat, 2.)

Dynaamisten web-sivujen lisäksi PHP:llä voi tehdä komentorivisovelluksia ja jopa graafisia käyttöliittymiä, mutta parhaiten sen ominaisuudet pääsevät esille web-sivuja kehiteltäessä.

Edellytyksenä PHP:n toiminnalle on palvelimen PHP-tuki, josta joutuu yleensä Suomessa maksamaan. Jos pelkästä PHP-tuesta maksaminen tuntuu epätaloudelliselta, niin Apache-palvelimen asentaminen Linuxiin kaikkine PHP-tukineen voi tällöin olla oikea ratkaisu. Tuen asentaminen omalle koneelle tarkoittaa yksinkertaisuudessaan siis sitä, että omasta koneesta tehdään täysin toimiva palvelin.

Palvelimen pystytys ei PHP:tä ohjelmoitaessa ole ollenkaan huono ajatus, sillä ilman tukea sivujen testaus kotikoneella ei onnistu, vaan sivut täytyy testausta varten siirtää aina palvelimelle, josta tuki löytyy. Tämä johtuu siitä, että PHP suoritetaan aina palvelimella. Tässä se eroaa ratkaisevasti monesta muusta kielestä, kuten Javascriptistä, joka suoritetaan käyttäjän selaimella. Koodin suorittaminen palvelimella tarkoittaa sitä, että käyttäjä ei näe sivua selatessaan koodia, vaan pelkästään suoritettua toimintaa tai lauseen.

Hyvä ja yksinkertainen esimerkki havainnollistamaan tätä tilannetta on mielestäni ohjelmoinnin kirjoittamattomiin sääntöihin kuuluva "Hello World!". Kyseisen lauseen on kirjoittanut varmasti jokainen ohjelmoinnin aloittelija aloittaessaan mitä tahansa kieltä.

Kuviosta 2 nähdään minkälainen kyseinen koodi on PHP:llä kirjoitettuna.

```
<?
echo "Hello World!";
?>
```

Kuvio 2. PHP yksinkertaisimmillaan.

Kun tämä upotetaan HTML-sivulle, niin käyttäjän selatessa sivun lähdekoodia, hän näkee vain tekstin "Hello World!", mutta ei aloitus- ja lopetustageja, eikä tulostuskomentoa 'echo'. Jos sivua katsoo ilman, että se on siirretty

palvelimelle, niin ruudulle tulostuu epäselvä merkkijono <?echo"Hello World";?>. PHP-tuen asentaminen omalle koneelle on mielestäni järkevää esimerkiksi modeemin tai kanssa työskenneltäessä, koska sivun siirto palvelimelle pienenkin muutoksen jälkeen vaatii aina yhteyden ottamisen, joten modeemin kanssa ohjelmointiin ja etenkin lopputulosten tarkistamiseen kuluu paljon aikaa.

PHP:stä on tarjolla useita eri versioita. Kaikkein uusin versio on PHP 5.3.2, joka julkaistiin maaliskuussa 2010. Tavallisesti ohjelmoija ei juurikaan huomaa eroa uusimpien kehitysversioiden välillä. Perusohjelmointi on siis aivan samanlaista, vaikka kyseessä ei olisikaan aivan uusin versio. PHP:n ensimmäisissä versioissa oli ilmestymishetkellä paljon aivan suoria virheitä, etenkin sessioita käytettäessä, mutta nyt ne on pääosin korjattu. Omaa palvelinta suunniteltaessa ja siinä käytettävää Linuxin versiota mietittäessä on syytä ottaa selvää etenkin vanhempien tai harvinaisempien Linuxien käyttämästä PHP:stä. Jotkut versiot voivat vaikuttaa ihan hyviltä palvelinympäristöiltä, mutta katsottaessa PHP:n versiota voi kyseessä olla esimerkiksi versio 1.0, jonka kanssa ohjelmointi ja käskyt saattavat jo erota uudemmista. Suosituimmat Linuxit tarjoavat kuitenkin poikkeuksetta toimivan version, joten esimerkiksi Ubuntu ja Fedora ovat käyttäjälle helppo valinta.

3 RELAATIOTIETOKANNAT

Relaatiotietokanta koostuu useista taulukoista. Taulukoissa tiedot esitetään riveillä ja sarakkeissa. Tietokannassa tarvittavat tiedot pyritään jakamaan taulukkoihin siten, että yksi tieto tallennetaan vain yhteen paikkaan. Relatiotietokantaan tallennetaan myös tieto siitä, miten eri taulukot liittyvät toisiinsa (Sarja, 5.). Relatiotietokannasta tietoa haetaan vain tiedon nimien ja arvojen perusteella, eikä koskaan esimerkiksi tiedon sijainnin tai järjestyksen mukaan.

3.1 Yleistä tietoa tietokannoista

Tietokanta on kokoelma tietoja, jotka ovat yhteydessä toisiinsa. Termiä tietokanta käytetään lähinnä tietotekniikassa, mutta esimerkiksi aivan tavallinen seinäkalenteri on tietokanta. Yrityksen asiakasrekisteriin voidaan koota tiedot asiakkaista ja näin saadaan tietokanta. Tietokannan osien välillä on oltava looginen yhteys. Jos yritys kirjaa asiakkaitaan ja heidän tietojaan tietokantaan, niin syötettäviksi arvoiksi voidaan pakottaa vaikkapa etunimi, sukunimi ja ammatti, eikä esimerkiksi pelkkää sukunimeä. Kun tietokannasta sitten haetaan joku asiakas esimerkiksi sukunimen ja ammatin perusteella, niin tietokannasta listataan täsmäävät arvot. Jos osumia on enemmän kuin yksi, voidaan hakua rajoittaa vielä lisää ja hakea myös asiakkaan etunimellä.

Tietotekniikassa tietokantojen koot voivat vaihdella aivan pienistä yhden taulukon tietokannoista jopa useisiin miljooniin tietueisiin. Tietokantoihin voidaan tallentaa tietoa monessa eri muodossa, kuten tekstinä, kuvina, ääninä tai videokuvina.

3.2 Käytännön sovelluksia tietokannoista

Tietokantoja käytetään monenlaisissa sovelluksissa, eikä niiden käyttö rajoitu pelkästään asiakasrekistereihin ja varaosakirjanpitoihin, vaan niitä voidaan käyttää millä sovellusalueella tahansa. Esimerkiksi monet sähköpostiohjelmat ja kalenterisovellukset perustuvat standardinmukaiseen tietokantateknologiaan. Yleisesti ottaen kaikenlaiset vuorovaikutteiset sovellukset, kuten palautelomakkeet, on erittäin kätevää toteuttaa tietokannoilla, puhumattakaan yritysten varaosaintitiedoista tai asiakasrekistereistä.

3.3 MySQL

Yksinkertaistetut tietokantayhteydet ovat varmasti yksi PHP:n parhaista ominaisuuksista. Monet tietokannat käyttävät kielenään SQL:ää. SQL on pienen totuttelun jälkeen varsin helppoa, sillä käskyt tulevat suoraan englannin kielestä.

PHP:ssä on valmiit funktiot tunnetuimmille ja yleisimmille tietokantaohjelmistoille, kuten MySQL:lle. (Järvenpää 2005, 1.)

Opinnäytetyössäni tutkin PHP:n ja MySQL:n ominaisuuksia tietokantoja käsiteltäessä. MySQL on helppokäyttöinen tietokanta ja yksinkertaisuutensa takia se soveltuu erittäin hyvin rajattuihin ja pieniin sovelluksiin. MySQL on ilmainen, eli esimerkiksi Linuxiin asennuspaketit voi ladata suoraan netistä. Toki ilmaisuus on rajoittanut MySQL:n piirteitä, joten todella monimutkaisiin sovelluksiin on syytä valita jokin maksullinen ja laajempi kanta.

MySQL, kuten PHP:kin, on siis ilmainen, mutta ilmaisuudesta on taaskin hyötyä vain, jos tuen voi asentaa omalle palvelimelle. Jos on turvaututtava Internetistä löytyviin kotisivutilan tarjoajiin, niin MySQL:ssä pätee sama kuin PHP:ssäkin, että Suomesta kyseisiä tukia ei helposti ilmaiseksi saa. Opinnäytetyöni yhtenä lähtökohdana on ilmaisuus, mutta PHP- ja MySQL tuesta Aatoskone Oy joutuu maksamaan.

Kun työskennellään MySQL:n, PHP:n ja Linuxiin asennetun palvelimen kanssa, voi törmätä termiin LAMP. Tällä tarkoitetaan alustaa, jolla työskennellään, eli Linux-Apache-MySQL-PHP. Yhteys PHP:n ja MySQL välille on helppo muodostaa. PHP koodiin kirjoitetaan yhteyskomennot (`mysql_connect`), joihin määritellään käyttäjätiedot. Sen jälkeen valitaan käytettävä tietokanta (`mysql_select_db`), ja sitten voidaan aloittaa hakukyselyiden suorittaminen. Yhteyttä otettaessa on syytä tehdä myös niin sanotut ”kuolemakomennot”, jotka antavat ilmoituksen, jos tietokantaan ei saadakaan yhteyttä. Näin on paljon helpompi lähteä hakemaan virhetilanteisiin ratkaisua. Palvelimen ollessa pois käytöstä tulisi ilmoitus, että palvelimeen ei saada yhteyttä ja virheellistä tietokantaa haettaessa, että kyseistä kantaa ei löydy. ”OR DIE” kannattaa siis kirjoittaa aina yhteyden ja tietokannan haun perään, koska ilman näitä on vaikeaa tietää, johtuuko jonkin sovelluksen toimimattomuus virheellisestä koodista, palvelimesta vai jostain muusta. Jos tietokannassa ei ole valmiiksi luotuja tauluja, niin nekin voidaan tehdä suoraan PHP:n avulla. Taulujen ja niihin tulevien sarakkeiden ja rivien muodostaminen on helppoa, koska käskyt

ovat yksinkertaisia, eivätkä ne sisällä mitään erikoismerkkejä tai muita kompastuskiviä.

Joidenkin todellisten ohjelmoijien mielestä on hankalaa ja hidasta käyttää MySQL –koodauksessa graafista hallintatyökalua phpMyAdmin. Selaimessa toimivaan hallintatyökaluun kirjaututaan sisälle, ja sen jälkeen tietokannoille voidaan tehdä, mitä halutaan. Paras tapa oppia ”mysliä” on varmasti phpMyAdministä löytyvä SQL-toiminto, jonka tekstikenttään voidaan suoraan kirjoittaa käskyjä ja sen jälkeen suorittaa ne. Käskyjen mennessä virheittä läpi tulee ilmoitus onnistuneesta lisäyksestä sekä koodinpätkä, joka on suoritettu. Jos ja kun koodissa on virheitä, niistä tulee virheilmoitus ja lyhyt kommentti, josta käy ilmi missä kohdassa ja minkä tyyppisestä virheestä on kysymys.

MySQL:ää opiskeltaessa yksi hyvä tapa on lähteä tutkimaan kieltä virhetilanteiden kautta. Kirjoittamalla yksinkertaiset käskyt tarkoituksella väärin pääsee hyvin alkuun siinä, minkälainen ilmoitus tulee aina mistäkin virheestä. Tietenkin on ihmisiä, jotka eivät halua kuluttaa aikaa SQL:n opiskelussa, joten heille phpMyAdmin tarjoaa helpommankin tavan luoda tauluja. Hallintatyökalun painikkeista saadaan nimittäin suoraan perustoiminnot, kuten juuri taulujen ja rivien lisäyksen. Kun lisätään vaikkapa valmiiseen tauluun uusi rivi, niin asia hoituu näppärästi pelkällä ”Lisää rivi” painikkeella. Tämä on toinen keino päästä alkuun kielen opiskelussa, sillä tässäkin tapauksessa onnistuneesti suoritettun käskyn näkee koodina sivulla. Toisin sanoen, jos lisätään yksi rivi suoraan valmiilla rivin lisäykseen tarkoitetulla painikkeella, niin toisen rivin lisäys voidaankin kirjoittaa jo itse suoraan SQL –laatikkoon, käyttäen apuna ohjatun toiminnon tarjoamaa koodia ensimmäisen rivin lisäyksestä.

3.4 Taulujen perustaminen MySQL:ään

Tietokantaan luodaan uusi taulu CREATE TABLE -komennolla. Taulun luonnin yhteydessä määritellään taulun kentät, niiden tietotyypit ja pituudet, sekä mahdollinen lisärajoite, kuten perusavain. Perusavain tekee jostakin kentästä ensisijaisen indeksin (PRIMARY KEY).

Kuvassa 3 tehtävä määrittely luo asiakas-taulun ilman minkäänlaista perusavainta. Kentät *etunimi* ja *sukunimi* ovat pakollisia eli niihin on syötettävä arvot. Tietotyypin jälkeen sulkujen sisällä on syötettävän arvon pituus.

```
CREATE TABLE asiakas
(
etunimi VARCHAR(20) NOT NULL,
sukunimi VARCHAR(20) NOT NULL,
puhelin NUMERIC (12)
);
```

Kuva 3. Taulun luonti MySQL:ssä.

Kun taulu on luotu, voidaan tehdä ensimmäinen rivin lisäys, eli syöttää ensimmäiset arvot kyseiseen tauluun. Tässä tapauksessa lisätään asiakkaan tiedot.

Kuvassa 4 rivin lisäys tehdään INSERT-komennolla:

```
INSERT INTO asiakas ( etunimi , sukunimi , puhelin )
VALUES ( 'Kalle' , 'Koodaaja' , '0700123123' )
```

Kuva 4. Arvojen lisäys tietokantaan.

Taulujen muokkaaminen, rivien poistaminen, tietojen päivitys ja kaikki muutkin toiminnot ovat suoraan englanninkielisiä käskyjä, joten pienet tietokannat pystytään tekemään muutamassa hetkessä vaikka silmät suljettuina, jos vain osataan vähänkin englantia.

Seuraavaan kuvioon on koottu SQL-92 standardin mukaisia tietotyyppejä:

TIETOTYYPPI	MERKITYS
CHAR	Kiinteän mittainen merkkijono.
VARCHAR	Vaihtuvanmittainen merkkijono.
NUMERIC	Numeerinen arvo
INTEGER	Kokonaisluku eli sanan pituus (4 tavua)
TIMESTAMP	Päiväys ja kellonaika

Kuvio 5. Yleisimmät tietotyypit.

4 SIVUJEN TOTEUTUS

Kun aloitetaan internetsivuston suunnittelu, mietitään tietenkin aluksi asiat ja sovellukset, joita halutaan esittää tai suorittaa. Kun aletaan puhua sivustoista, jotka käyttävät tietokantoja, niin ensimmäisenä tulee tehdä valinta siitä, mitä kieliä ja tietokantoja käytetään.

4.1 Työskentely MySQL-PHP -yhdistelmällä

PHP-MySQL on koko ajan yleistynyt ja suosiotaan nostava yhdistelmä, joten ohjelmointialustan valinta oli helppoa. PHP:n korvaaminen Perlillä olisi ollut mielestäni askel taaksepäin, sillä tämän tyyppisiä tietokantoja (web-sivut) käsiteltäessä PHP on ylivoimainen. Perlillä on enemmänkin tarkoitus lukea läpi tekstitiedostoja, poimia niistä tietoa ja tulostaa niistä tältä pohjalta raportteja. Perlin hyvänä ominaisuutena pidän sitä, että se ei aseta rajoituksia datan koolle, joten jos muistia on käytössä tarpeeksi, Perl voi napata koko tiedoston yhteen ainoaan merkkijonoon. Aatoskone Oy:n varaosakirjanpidon kanssa pyöritellään kuitenkin niin pieniä tietoja, että olisi ollut aivan turha lähteä työskentelemään itselleni aivan tuntemattoman kielen, Perlin, kanssa.

4.2 Käytetyt ohjelmat

Yhtenä lähtökohdiana oli tehdä kaikki mahdollisimman pienin kustannuksin. Tarkoituksena oli käyttää vain ilmaisia ohjelmistoja. Olen tehnyt sivuja

pääasiassa kahdella editorilla, ensin suomalaisella SuperHTML Padilla ja sen jälkeen jo useamman vuoden Crimson Editorilla. Crimson Editor on ilmainen tekstieditori, joka tukee kaikkia yleisimpiä ohjelmointikieliä, joten ohjelmointi on selkeää, koska koodin näkee väreinä ja esimerkiksi kirjoitusvirheet on näin ollen helpompi löytää ja korjata.

Aatoskone Oy:n verkko-operaattorin palvelimella on käytettävissä phpMyAdmin, joten SQL-koodauksen sai tehtyä mukavasti graafisessa ympäristössä, eikä tarvinnut kirjoittaa käskyjä PHP:n sekaan. FTP –ohjelmana, jolla sivut siirretään palvelimelle, käytin FlashfXP:tä, jonka valitsin siitä syystä, että sillä on mahdollista muokata jo palvelimellekin siirrettyjä sivuja. Pienten muutosten tekeminen oli erittäin nopeaa, koska ei tarvinnut aina siirtää omalta koneelta kokonaista tiedostoa uudestaan. Toinen erittäin hyvä ominaisuus FlashfXP:ssä on se, että sillä voidaan määritellä tiedostojen oikeudet (CHMOD). Jos halutaan joillekin sivuille vaikkapa kirjautuminen, niin esimerkiksi salasana tiedot sisältävä sivu saadaan piiloon kätevästi pelkillä CHMOD –oikeuksien muuttamisella. FlashfXP ei ole ilmainen, mutta muutaman kuukauden kokeilu-aika riitti juuri ja juuri opinnäytetyöni tekemiseen.

Sivujen testaus tapahtui aina Mozilla Firefoxilla, koska Microsoftin Internet Explorerissa on koodaajien kannalta yksi todella rasittava ja epäkäytännöllinen ominaisuus. Explorer nimittäin ymmärtää jonkun verran virheellistäkin koodia, joten jos sivu näyttää Explorerissa hyvältä ja toimivalta, niin se ei takaa vielä mitään.

4.3 Työn kulku

Varaosakirjanpitosovellusta lähdin toteuttamaan siltä pohjalta, että yksi varaosa olisi aina yksi lisäys suoraan tietokantaan. Toinen vaihtoehto olisi ollut, että varaosa tallentuisi aina suoraan tekstitiedostoon, mutta näin ollen MySQL olisi jäänyt kokonaan pois.

Aloitin varaosakirjanpitosovelluksen teon siten, että ensin tutustuin phpMyAdminin toimintaan. Kun kyseisen hallintatyökalun avulla MySQL:n

perustoiminnot alkoivat hahmottua, pystyin perustamaan tietokantaan Varaosa-
taulun. Ennen kuin tein PHP:n kanssa vielä yhtään mitään, täytyi Varaosa
taulun sarakkeisiin lisätä sellaiset kohdat, joita varaosasovellusta käyttävältä
henkilöltä kysytään. Sarakkeita tauluun tuli viisi ja ne ovat aivan perustietoja, eli
tuote, varaosnumero, määrä, huomioitavaa ja tilaajan kuittaus. Lisäksi tein
päivämäärän TIMESTAMP-tietotyypillä ja se lisätään automaattisesti jokaiseen
varaosaan. TIMESTAMPin päivämäärä on amerikkalaisessa muodossa ja sen
saaminen selkeämpään muotoon oli erittäin hankalaa.

Kun sain taulun ja sarakkeet luotua, olin valmis aloittamaan PHP:llä
ohjelmoimisen. Otin varaosakirjanpidon periaatteen aikaisemmin tekemästäni
vieraskirjasovelluksesta, jossa voidaan päivittää, poistaa ja lisätä vieraskirjan
viestejä tietoja. Varaosakirjanpitosovelluksen ideahan on melkein samanlainen
kuin vieraskirjan viestin lisäys, eli taulussa oleviin sarakkeisiin lisätään annetut
arvot. Varaosakirjanpitosovelluksen runko olikin itse asiassa melko
yksinkertainen toteuttaa, sillä vieraskirjatiokannasta sain todella hyvän mallin
ja paljon apua. Varaosakirjanpitosovelluksen PHP-koodi tarvitsee toimiakseen
kaksi sivua. Ensimmäisellä sivulla on HTML -lomake, jossa pyydetään
lisäämään varaosa ja täyttämään kysytyt kohdat (tuote, määrä jne.).
Ensimmäisen sivun PHP -koodissa otetaan standardin mukainen yhteys
tietokantaan, suoritetaan hakukysely Varaosa-taulusta ja haetaan kaikkien
sarakkeiden arvot. Tämä tehdään, jotta nähdään tällä sivulla lisäyslomakkeen
alapuolella myös kaikki aiemmin lisätyt varaosat. Näin varaosille ei tarvitse
tehdä kokonaan omaa sivua. Lomakkeen alalaidassa on painike "Lisää".
Painikkeen painallus ohjaa suoraan sivulle, jossa varsinainen lisäys tehdään.
Kun täytetään lomake ja painetaan Lisää-painiketta, ohjautumisen jälkeen
avataan ensin yhteys tietokantaan ja tarkistetaan sitten se, onko tuote ja määrä
-kentissä arvoja. Jos tuote tai määrä ovat tyhjiä, ei lisäystä tehdä, vaan
palataan takaisin sivulle, jossa painiketta painettiin.

Valintarakenne, joka tarkistaa tyhjät lisäykset:

```
if ($tuote!=""&&$maara!="") (1)
```

Jos lomakkeesta löytyvät tarvittavat tiedot, jatkaa PHP eteenpäin. INSERT-komennolla lisätään kaikki annetut arvot tauluun, ja jos tietokanta on kunnossa, tuleeekin tämän jälkeen ilmoitus onnistuneesta varaosan lisäyksestä (INSERT-komennosta lisää luvussa 3.4). Vikatilanteissa ei tehdä lisäystä, vaan ilmoitetaan virheestä. Virheellinen lisäys tulee silloin, kun palvelin ei vastaa tai palvelimen omissa MySQL-yhteyksissä ja kannoissa on jotain vialla. Tein tälle sivulle HTML:llä uudelleen ohjauksen, joten käyttäjän ei tarvitse varaosan lisäyksen jälkeen painaa mistään, vaan selain siirtyy automaattisesti edelliselle sivulle, jossa näkyikin juuri lisätty varaosa muiden varaosien joukossa. Seuraavassa kuvassa varaosakirjanpidon toimintaperiaate.



Kuva 6. Varaosakirjanpidon toiminnan perusajatukset.

4.4 Varaosakirjanpidon ominaisuudet

MySQL valittiin tekstitiedoston sijasta sovelluksen pohjaksi seuraavien asioiden takia. Varaosien lisäys olisi toiminut yhtä lailla niin tekstitiedostoon kuin MySQL:ään, mutta kun lähdetään miettimään tilannetta, jossa väärät tai varaosasta poistuneet tuotteet haluttaisiin poistaa, niin MySQL osoittaa paremmuutensa. MySQL:n hallintatyökalusta (phpMyAdmin) varaosa saadaan pois parilla hiiren klikkauksella, mutta tekstitiedostosta poistaminen olisi ollut aivan liian hankalaa ja käytännössä lähes mahdotonta.

Vaikka varaosien poisto onnistuu suoraan phpMyAdminista, niin halusin tehdä kuitenkin niin sanotun hallintasivun, jonne kirjautumalla voisi poistaa varaosia. Tämä hallintasisu tehtiin siitä syystä, että varaosia tilataan kuitenkin päivittäin ja tilauksia täyttää koko yrityksen henkilöstö. Joten jos virheellisiä varaosia on kirjattu paljon, niin ne voidaan poistaa juuri siihen tarkoitukseen tehdyllä sivulla,

kuin phpMyAdminilla, joka on kuitenkin loppujen lopuksi tarkoitettu enemmänkin kantojen perustamiseen ja taulujen ja niissä olevien kenttien päivitykseen.

Yksi ehdottoman tärkeä ominaisuus, jonka halusin sisällyttää varaosakirjanpitoon, oli käyttäjätietojen keräys. Käyttäjätietojen keräys on aiheellista, koska vuosittain tehtävässä inventaariossa paljastuu aina varaosia, joita kukaan ei myönnä tilanneensa. Käyttäjätiedot keräämällä on helppoa lähteä selvittämään, kuka on tilannut ja mitä. Teinkin sivuille koodinpätkän PHP:llä, joka kerää Lisää-painiketta painaneen käyttäjän IP-osoitteen. Aatoskone Oy käyttää omaa sisäverkkoa, jossa on kiinteät IP-osoitteet, joten varaosakirjanpidosta näkee suoraan kenen tietokoneelta varaosat on tilattu.

Näin haetaan PHP:llä IP-osoite:

```
$REMOTE_ADDR;      (2)
```

Tällä kertaa tekstitiedosto oli parempi vaihtoehto tietojen tallentamiseen, sillä näin lyhyeen koodin olisi ollut kyseenalaista alkaa kirjoittaa yhteyden hakuja ja lisäyskomentoja. IP-osoitteen seuraksi olisi ollut mahdollista ottaa myös selaimen tiedot ylös, mutta tällä tiedollahan yritys ei tee yhtään mitään. Kaiken lisäksi PHP ei ymmärrä alkuunkaan sitä, minkälainen käyttäjän selain on. Tämä johtunee siitä, että selaimen tiedot keräävä funktio on kehitetty PHP:n ensimmäisiin versioihin, jolloin käytössä ei ollut kuin pari toimivaa internetselainta.

Käyttäjän IP-osoitteen kerääminen ei ole täysin varma tapa selvittää varaosien tilaajaa. Työntekijä ei välttämättä tilaa varaosia omalta tietokoneeltaan tai hän voi tilata varaosat esimerkiksi kotoaan. Tämän takia varaosatilausta täytettäessä käyttäjän on lisättävä myös oma nimensä.

Käyttäjätietojen lisäksi halusin varaosakirjanpitoon ominaisuuden, joka estää sanojen tulvimisen, eli "floodauksen". Floodaus tarkoittaa sitä, että joku haluaa olla hauska ja painaa näppäimistön näppäimet pohjaan ja kenttään tulostuu kymmenentuhatta merkkiä. Tällainen tulva aiheuttaa sen, että sivun ulkoasu

menee sekaisin. Tähän ongelmaan etsin ratkaisua jonkin aikaa ja viimein PHP:n omilta websivuilta (<http://www.php.net>) löytyi tähän soveltuva funktio, wordwrap.

Wordwrapin idea on siinä, että se sisällytetään johonkin muuttujaan ja kun muuttuja saa arvon, tarkistetaan, onko merkkijono liian pitkä. Jos merkkijono ylittää sallitun rajan, esimerkiksi 60 merkkiä, wordwrap pilkkoo ja rivittää sanan uudelleen. Näin sivujen ulkoasu pysyy siistinä, eikä leviä epämääräisesti ympäri ruutua. Käytin kyseistä apukeinoa *huomioita*-kentässä.

Huomioita-kohdan ongelmana oli pitkään se, että käytin huomioita – sarakkeessa tietotyyppiä INTEGER, jonka enimmäispituus on 255 merkkiä. Noin lyhyellä merkkijonolla ei kirjoiteta kuin välttämättömin, joten aloin miettiä, millä saisin huomioita -kentän pituuden suuremmaksi. Kokeilin syöttää suurempaa lukua kuin 255, mutta MySQL antoi virheilmoituksen. Sattumalla kiinnitin kuitenkin virheilmoituksen pienellä kirjoitettuun tekstiin huomiota, ja siellä mainittiin jotain sellaisesta tietotyyppistä kuin BLOB. BLOB on tarkoitettu lähinnä kuvien käyttöön, mutta kokeilunhaluisena vaihdoin sen kuitenkin käyttöön. PhpMyAdmin osoitti hyödyllisyytensä, sillä BLOB-funktio alkoi toimia saman tien, ja nyt huomioita-kenttään voi kirjoittaa rajattoman määrän merkkejä. Huomioita –kenttään voi nyt esimerkiksi kirjoittaa jälkitoimitukseen, tai rahtiin liittyvistä seikoista.

4.5 Suojattu hallintasivu

Koska halusin tehdä varaosakirjanpidolle hallintasivun, täytyi myös alkaa miettiä salasanasuojausta, koska ilman salasanaa kuka tahansa voisi poistaa varaosia. Ensimmäinen ajatus minulla oli, että tekisin myös kirjautumisen MySQL:llä. Omat ohjelmointitaitoni ovat kuitenkin vielä kaukana siitä, mitä tarvittaisiin toimivan kirjautumisjärjestelmän tekoon. Internetistä sain toimivan MySQL rekisteröitymisen, mutta sen muokkaaminen omille sivuille sopivaksi oli näillä taidoilla niin hidas urakka, että aika loppui kesken. Toinen syy, joka asettui MySQL:ää vastaan, oli se, että kukaan muu ei tarvitsisi salasanaa kuin

yrittäjien johto. Pelkän yhden pienen tiedon hakeminen MySQL:stä oli siis mielestäni myös resurssien haaskausta.

Aloin selata läpi ohjelmointifoorumeita ja etsiä toimivaa kirjautumista, jossa salasana tiedot olisivat tekstitiedostossa. Kokeilin ja muokkasin varmaan puoli tusinaa erilaista koodia. Lopulta löytyikin yksi, jonka sain pienellä viillauksella ohjelmoitua itselleni sopivaksi. Alkuperäisen koodin on tehnyt nimimerkki *Tumpi* ja se on julkaistu 1.5.2003 osoitteessa <http://www.ohjelmointiputka.net>.

Kirjautuminen perustuu sessioihin, eli siihen, että käyttäjän sessio (istunto) kestää, kunnes se lopetetaan ulos kirjautumisella tai selaimen sulkemisella. Salasanatiedot ajetaan MD5:een, jota käytetään tietojen autentikoimiseen, eli siihen että tehdään niistä luotettavia ja salattuja. MD5 luo 128 bitin tiivisteen jostakin sanasta tai merkkijonosta, joka tyypillisesti esitetään 32-merkkisenä heksakoodatussa muodossa. Laskennallisesti on mahdotonta löytää toista samanlaista tietoa, joten vaikka salasanat ovatkin palvelimella luettavissa suoraan tekstitiedostosta, niin niitä ei pysty purkamaan.

Toki on arveluttavaa pitää salattujakaan tietoja esillä, joten jos on kysymys vähänkin tärkeämmästä sovelluksesta kuin varaosakirjanpidosta, niin kannattaa ehdottomasti miettiä parempia ja luotettavampia ratkaisuja. Tässä tapauksessa kyseinen menetelmä on kuitenkin ihan riittävä. Kaiken lisäksi opinnäytetyöni on vasta sovelluksen ensimmäinen kehitysversio, jota tulen jatkossa päivittämään kehittyneemmäksi. Tällä hetkellä varaosakirjanpitosovellusta käytetään ainoastaan niin sanottuna tilauskirjana.

Sovellus ei esimerkiksi huomio vielä ollenkaan varaosien kappalemääriä, eli jos jotain tuotetta on useita niiden varaosien saldot eivät muutu tilausta tehtäessä. Sovellus kerää siis ainoastaan listaa, johon työntekijät merkitsevät tilaamansa varaosat. Varaosakirjanpitosovellukseen merkityt tuotteet käydään kerran kuukaudessa läpi ja ne tulostetaan arkistoitavaksi. Jos tiedot jostain syystä häviäisivät, ei tapahdu mitään vakavaa onnettomuutta, koska kaikesta löytyy aina vähintään kuukauden takainen tilanne myös paperiversiona.

Rekisteröin itselleni yhden käyttäjätunnuksen ja poistin sen jälkeen koko sivun, jotta kukaan muu ei voi rekisteröityä. Sen jälkeen liitin varaosakirjanpidon hallintasivulle PHP includen, joka vaatii sen, että käyttäjätunnus on pitänyt syöttää onnistuneesti ennen kuin sivulle pääsee. Jos sivulle yrittää mennä ilman kirjautumista, kysytään salasanaa.

Varaosan poisto on hyvin samanlainen operaatio kuin varaosan lisäyskin. Lähtökohtana on, että INSERT korvataan DELETELLÄ. Hallintasivu on sama kuin se sivu, jolla varaosia lisätään, mutta jokaisen valmiin varaosan perässä on Poista-painike. Painikkeen painaminen ohjaa tässäkin tapauksessa uudelle sivulle, mutta tällä kertaa varaosaa ei lisätä vaan poistetaan.

4.6 Sivuston ulkoasu

Kun olin saanut varaosakirjanpidon toimimaan haluamallani tavalla, saatoin aloittaa ulkoasun tekemisen. World Wide Web eli WWW kehiteltiin vuonna 1990, ja hanke oli puhtaasti tieteellinen. Verkon kautta levitettiin ainoastaan tieteellisiä raportteja, joiden sisältö oli paljon ulkoasua tärkeämpää. WWW:n saama valtaisa suosio pakotti kuitenkin miettimään keinoja, joilla internetsivustot saataisiin käyttäjäystävällisiksi. Tämän takia on kehitelty tyylikieli CSS. CSS perustuu sääntöihin ja tyylimäärittelyihin. Sääntö on yhden tai useamman elementin tyylimäärittely. HTML-sivujen muotoilussa tyylimäärittely muodostuu yhdestä tai useammasta säännöstä, joita selain käyttää. Tyylimäärittelyt toimivat samalla tavalla kaikissa uudemmissa selaimissa.

Yksinkertainen tyylimäärittely sivun otsikolle H1:

```
H1 { color : blue }      (3)
```

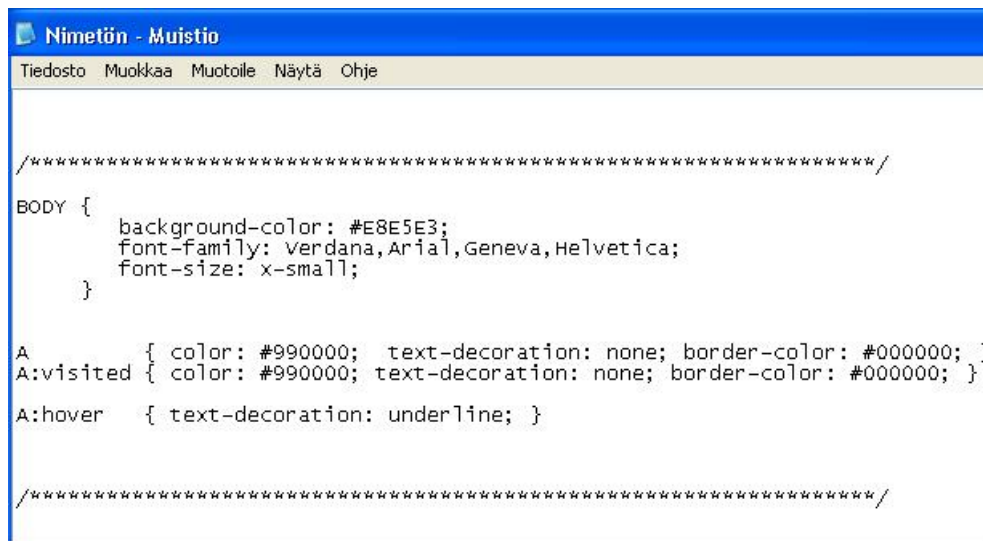
Sivuston ulkoasun tyylimäärittelyt halusin selkeyden takia sisällyttää kokonaan erilliseen tiedostoon. Näin sivuston lähdekoodi kestää nopeana ja helppolukuisena.

CSS otetaan käyttöön seuraavalla määrittelyllä:

```
<LINK REL=STYLESHEET HREF="tyyli.css"> (4)
```

Ulkoasusta tein sellaisen, että se pysyy selkeänä, vaikka sitä katsoo millä resoluutiolla tahansa. Kun taulukoiden koot ilmoittaa prosentteina, niin taulukot sulautuvat mille näytön resoluutiolle tahansa. Jos taulukon koko ilmoitetaan pikseleinä, niin leveys on aina sama, vaikka sivua katsoisi matkapuhelimella. Taulukoilla ja tyylimäärittelyillä sain aikaan käyttäjäystävällisen ulkoasun. Myös sivuston päivittäminen on paljon helpompaa, koska ei tarvitse muuttaa jokaisen sivun ulkoasua erikseen, vaan vaihtamalla esimerkiksi sivuston taustaväriin CSS-tiedostosta, vaihtuu väri jokaisella sivulla.

Seuraavassa kuvassa tehdään sivun värien, tekstien, sekä linkkien määrittäykset:



```
Nimeton - Muistio
Tiedosto Muokkaa Muotoile Näytä Ohje

/*****/
BODY {
    background-color: #E8E5E3;
    font-family: Verdana,Arial,Helvetica;
    font-size: x-small;
}

A      { color: #990000; text-decoration: none; border-color: #000000; }
A:visited { color: #990000; text-decoration: none; border-color: #000000; }
A:hover  { text-decoration: underline; }

/*****/
```

Kuva 7. CSS-tyylimäärittelyt.

Kun sain sivujen ulkoasun valmiiksi, oli Aatoskone Oy valmis ottamaan varaosakirjanpitosovelluksen käyttöön.

5 YHTEENVETO

Tähän projektiin lähteminen oli sekä tuttua että turvallista, mutta myös toisaalta hyppy tuntemattomaan. Olen ollut HTML:n kanssa tekemisissä jo useita vuosia, ja tavallisten web-sivujen osaaminen antoi rohkeutta tarttua minulle täysin vieraaseen MySQL:ään. PHP:tä olin joskus yrittänyt kokeilla, mutta heikoin tuloksin. PHP:n kanssa työskentely ei ollut läheskään niin haastavaa kuin olin ennakolta ajatellut. Ohjelmointikielet ovat olleet minusta aina mielenkiintoisia, mutta C ja Javascript ovat opettaneet minut arkailemaan ohjelmoinnin kanssa. Syy on varmasti ollut siinä, että en ole heti tajunnut esimerkiksi C-kielen peruseriaatteita ja liian monet virheelliset ohjelmointirytykset ja toimimattomat lauseet ovat synnyttäneet pelkoa ja ennako-odotuksia ohjelmointikieliä kohtaan. Tästä syystä PHP:n selkeys yllätti minut. Myös internetistä löytyvä materiaali on niin laajaa, että ongelmien ratkaisu onnistui yleensä pelkällä keskustelufoorumien tai koodikirjastojen tutkimisella.

MySQL:n yksinkertaiset toiminnot ja tietokantayhteydet PHP:hen hahmottuivat minulle ihan kohtalaisesti, sillä asian opiskeluun ei kulunut kuin pari kuukautta. Tämän mittaisen opiskelun ja harjoittelun tuloksena yllätyin siitä, miten hyvin sisäistin juuri kaiken perustietouden, jota jouduin opinnäytetyössäni käyttämään.

Työtä voi lähteä helposti kehittämään eteenpäin. Taulujen välisiä yhteyksiä työssäni ei ole, joten niiden rakentaminen onkin seuraava haaste. Lisäksi sovelluksen tulee jatkossa pystyä päivittämään varaosien määrät aina, kun uusi osa tilataan tai myydään.

Nyt olen kuitenkin luonut pohjan yrityksen varaosakirjanpidolle, mutta lisäominaisuuksien kehittäminen luo minulle jatkossakin haastetta.

Opinnäytetyöni yksi ehdottoman tärkeä periaate oli minimoida kustannukset. Vastaavia, ja tottakai paljon kehittyneempiä, sovelluksia on tarjolla vaikka kuinka paljon, mutta yritys halusi yksinkertaisen ja helppokäyttöisen sovelluksen mahdollisimman halvalla. Mielestäni tässä onnistuttiin vähintäänkin

kohtalaisesti, sillä opinnäytetyönäni tekemä sovellus on kuitenkin yrityksen aiempaa järjestelmää (ruutuvihko ja kuivamustekynä) huomattavasti selkeämpi ja helppokäyttöisempi.

LÄHTEET

Järvenpää, J-P 2005.

<http://www.sivut.org/> (luettu 3.2010-5.2010)

KK Mediat.

<http://www.2kmediat.com/php/> (luettu 3.2010-5.2010)

MySQL

<http://www.mysql.com> (luettu 3.2010-5.2010)

PHP

<http://www.php.net> (luettu 3.2010-5.2010)

Sarja, J 2006.

<http://www.verkkopedagogi.net/> (luettu 19.5.2010)

Wikipedia: WWW-sisällönhallinta

<http://www.wikipedia.fi> (luettu 22.5.2010)

Sivu, jolla näytetään jo tilatut osat, sekä täytetään tilauskaavake.

```
<?
    //Avataan yhteys tietokantapalvelimeen
    mysql_connect("palvelin","käyttäjätunnus","salasana") or die ("Yhdistäminen ei onnistunut!");
    mysql_select_db("Varaosat") or die ("Tietokantaa ei löytynyt");

    //Tehdään hakukysely
    $kys = "select*FROM Varaosat ORDER BY huomioID";
    $tulos=mysql_query($kys);

        {
            echo "<TD><P class=/perus/><b>Tuote: </b>". mysql_result ($tulos,
    $i,"tuote")."</font></p>";
            echo "</TD>";

            echo "<TR><TD><P class=/perus/><b>Varaosanumero: </b>". mysql_result
    ($tulos,$i,"varaosanumero")."</font></p>";
            echo "</TD></TR>";

            echo "<TR><TD><P class=/perus/><b>Määrä: </b>". mysql_result ($tulos,
    $i,"maara")."</font></p>";
            echo "</TD></TR>";

            echo "<TR><TD><P class=/perus/><b>Huomio: </b><BR>";
            $text = mysql_result ($tulos,$i,"huomio");
            $newtext = wordwrap($text, 80, "<br />\n")."</font></p>";
            echo $newtext;
            echo "</TD></TR>";

            //Päivämäärä selkeään muotoon
            echo "<TR><TD><P class=/perus/>";
            ereg("([0-9]{4})([0-9]{2})([0-9]{2})([0-9]{2})([0-9]{2})", mysql_result($tulos,
    $i,"date"), $regs);

            echo "<BR>". $regs[3].".".$regs[2].".".$regs[1].".".$regs[4].".".$regs[5];
            echo "</TD></TR>";
            echo "<TR><TD><HR></TD></TR></p>";

        }

    mysql_close();

?>
```

Sivu, joka lisää täytetyt kentät tietokantaan.

```
<html>
<head>
<meta http-equiv="refresh" content="1;url=./varaosat.php">

<title></title>
</head>
<body>

<?
//Avataan yhteys tietokantapalvelimeen
mysql_connect("palvelin","käyttäjätunnus","salasana") or die ("Yhdistäminen ei onnistunut!");
mysql_select_db("Varaosat") or die ("Tietokantaa ei löytynyt");

//Tarkistetaan onko tuote- ja huomio kentissä arvot
if ($tuote!=""&&$huomio!="")

{
    //Lisätään tiedot tietokantaan
    $kys="INSERT INTO Varaosat (tuote, varaosanumero, maara, huomio)
VALUES ('$tuote', '$varaosanumero', '$maara', '$huomio)";

    $tulos=mysql_query($kys);

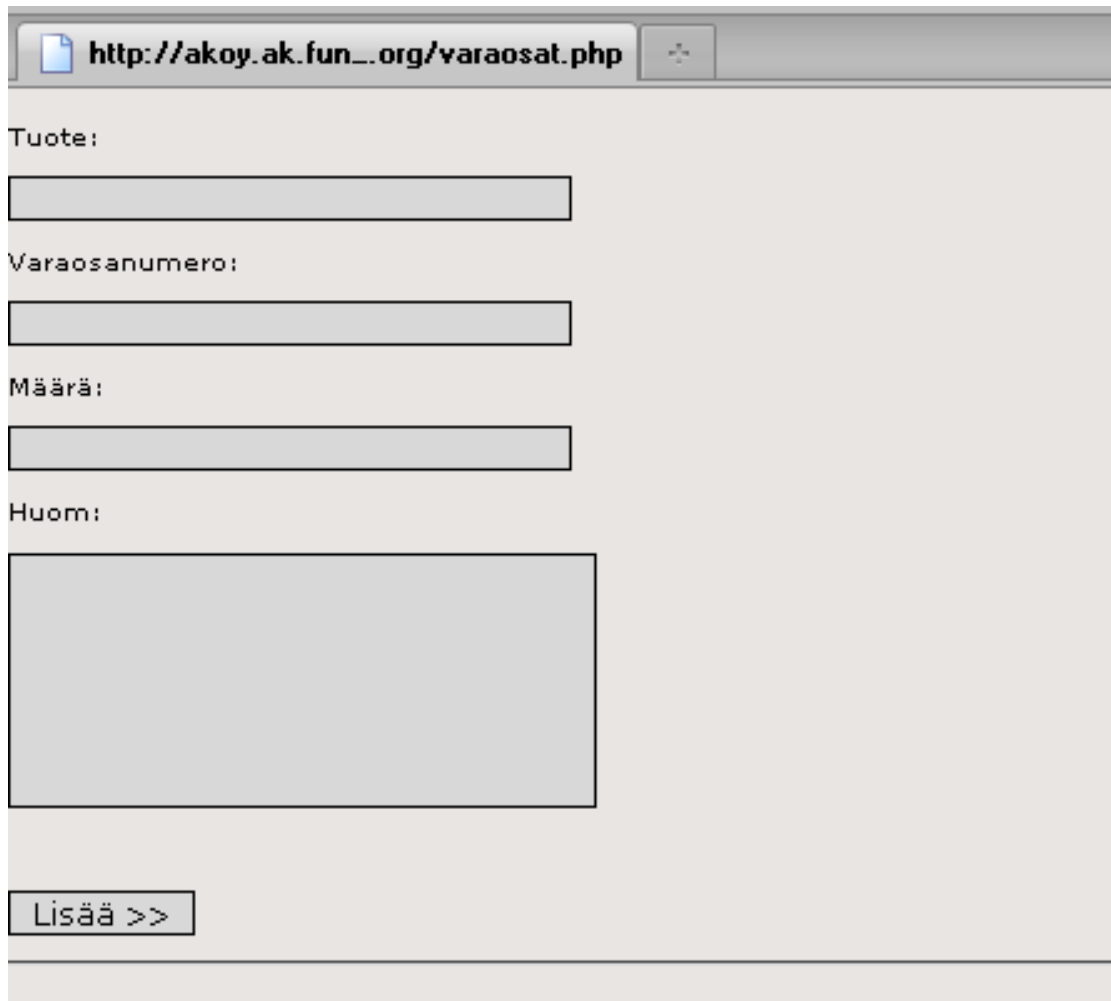
    if ($tulos==0)
        echo "Lisäys epäonnistui!";
    else
        echo "Lisäys onnistui!";
}

mysql_close();

?>

</body>
</html>
```

Kuva valmiista varaosakirjanpitosovelluksesta.



The image shows a screenshot of a web browser window. The address bar contains the URL `http://akoy.ak.fun...org/varaosat.php`. The main content area of the browser displays a form with the following fields and labels:

- Tuote:** A text input field.
- Varaosanumero:** A text input field.
- Määrä:** A text input field.
- Huom:** A large text area for notes.

At the bottom of the form, there is a button labeled **Lisää >>**.