Matti Vähänen

VERKKOMARKKINOINNIN TEKNINEN SUUNNITTELU JA TOTEUTUS

Esimerkkinä TT Sisustus tmi

Opinnäytetyö
CENTRIA AMMATTIKORKEAKOULU
Mediatekniikan koulutusohjelma
Kesäkuu 2014

TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Yksikkö	**Aika**	**Tekijä/tekijät**
Ylivieska	Kesäkuu 2014	Matti Vähänen
Koulutusohjelma Mediatekniikka		
Työn nimi VERKKOMARKKINOINNIN TEKNINEN SUUNNITTELU JA TOTEUTUS. Esimerkkinä TT Sisustus tmi		
Työn ohjaaja Timo Taari		**Sivumäärä** 25+1
Työelämäohjaaja Tarja Tuulia Leppälä		

Opinnäytetyön aiheena oli suunnitella ja toteuttaa web-sivuston palvelinohjelmointi ja sisällönhallintajärjestelmä. Toteutus sisälsi kauppapaikan tuotteiden ja niihin liittyvien tietojen hallinnan ja noudon tietokannasta sekä sivuston sisällönhallintajärjestelmän.

Asiasanat
Internet, JavaScript, kuvamanipulaatio, kyselykielet, MySQL, ohjelmistosuunnittelu, PHP, relaatiotietokannat, salasanat, sisällönhallinta, verkkoliiketoiminta, verkko-ohjelmointi, www-sivustot

ABSTRACT

CENTRIA UNIVERSITY OF APPLIED SCIENCES Ylivieska	Date June 2014	Author Matti Vähänen
Degree programme Media Technology		
Name of thesis TECHNICAL DESIGNING AND EXECUTING OF ONLINE MARKETING. Case: TT Sisustus tmi		
Instructor Timo Taari		**Pages** 25+1
Supervisor Tarja Tuulia Leppälä		

The topic of this thesis was to design and create programming of web server and content management system. The implementation included managing and fetching products and their information from database. The implementation contained also content management system for the website.

Key words
Content management, Internet, JavaScript, MySQL, online business, passwords, photo manipulation, PHP, query languages, relational databases, software design, web programming, websites

KÄSITTEIDEN MÄÄRITTELY

Ajax (Asynchronous JavaScript and XML)
Nimitys web-sivun vuorovaikutteisuuden mahdollistavasta JavaScriptin ja XML:n yhteiskäytöstä.

Alfakanava
PNG-formaatiin RGB-värikanavien lisäksi sisältyvä neljäs värikanava.
Mahdollistaa asteittaisen läpinäkyvyyden.

BMP (Bitmap)
Pakkaamaton tiedostomuoto bittikarttakuvien tallennukseen.

DOM (Document Object Model)
Menetelmä web-sivun rakenteen käsittelemiseksi hierarkisena oliona.

Exif (Exchangeable image file format)
Muun muassa JPEG-kuvissa käytetty menetelmä metatietojen tallentamiseksi.
Sisältää tietoja kuvasta, kuvan ottaneesta laitteesta ja kuvausasetuksista.

GIF (Graphics Interchange Format)
Nykyään jo vanhentunut tiedostomuoto piirroskuvien tallentamiseen. Rajallinen värien määrä (256).

HTML (Hypertext Markup Language)
Sivunkuvauskieli, jolla kuvataan www-sivuja.

HTTP (Hypertext Transport Protocol)
Protokolla tiedostojen siirtoon palvelimen ja selaimen välillä.

JavaScript

Selaimessa suoritettava dynaamisten web-sivujen ohjelmointikieli.

JPEG (Joint Photographic Experts Group)

Yleisesti käytetty tiedostomuoto valokuvien tallentamiseen.

MIME-tyyppi (Internet Media Type)

Alun perin sähköpostijärjestelmiä varten luotu menetelmä tiedoston tyypin tunnistamiseksi.

Palvelin

Verkkoon kytketty erityinen tietokone, joka suorittaa toisille tietokoneohjelmille palveluita tarjoavia ohjelmistoja.

PHP (PHP: Hypertext Preprocessor)

Palvelimella ajettava komentotulkki

PNG (Portable Network Graphics)

Tiedostomuoto erityisesti piirroskuvien tallentamiseen ja esittämiseen verkossa.

RGB (Red Green Blue)

Värijärjestelmä, jonka eri värit muodostuvat punaisen, vihreän ja sinisen valon sekoituksista.

Selain

Ohjelma, joka tulkkaa sisällön HTML-kielestä esitysmuotoon eli näkyviksi web-sivuiksi.

Skripti

Komentosarja, jolla toteutetaan yksinkertaisia toimintoja tietokoneessa. Ei varsinaisesti ohjelmointikieli.

SQL (Structured Query Language)

Useimpien tietokantaohjelmistojen käyttämä standardoitu kyselykieli.

Suola

Salattavaan merkkijonoon liitettävä merkkijono, joista muodostuneesta merkkijonosta luodaan tiiviste salausalgoritmillä.

URI (Uniform Resource Identifier)

Merkkijono, joka ilmaisee dokumentin polun, sijainnin palvelimella. Esimerkiksi /index.php

URL (Uniform Resource Locator)

Merkkijono, joka ilmaisee dokumentin sijainnin globaalisti. Sisältää käytettävän protokollan, palvelimen osoitteen ja dokumentin polun. Esimerkiksi http://ttsisustus.fi/index.php

XML (Extensible Markup Language)

HTML:ää muistuttava merkintäkieli, joka sisältää kuvauksen sisältämistään tietokentistä.

1 JOHDANTO

TT Sisustus on loppuvuodesta 2010 perustettu sisustusalan toiminimi. Toiminimi työllistää yhden henkilön ja yrityksen liiketoiminta on kasvanut perustamisesta lähtien. Toimittuaan noin kaksi vuotta toiminimelle syntyi tarve omille web-sivuille ja sähköiselle kauppapaikalle. Sähköisen kauppapaikan tavoitteena on esitellä toiminimen tuotevalikoimaa ja tarjota asiakkaille yhteydenottokeino kauppapaikan yrittäjään.

Sähköinen kauppapaikka on hieman erilainen kuin perinteinen verkkokauppa: sähköinen kauppapaikka on verkkokaupan ja tavallisten web-sivujen välimuoto, jossa yrityksen tuotteet ja palvelut ovat esillä, mutta niitä ei voi tilata sieltä suoraan. Tämä on ymmärrettävää, sillä TT Sisustuksen tapauksessa tuotteet ovat yksilöllisiä ja ainutlaatuisia eikä niitä valmisteta suuria eriä eikä niitä siten ole varastossa läjäpäin. Tämä antaa myös asiakkaalle mahdollisuuden tuotteiden personointiin.

Kauppapaikan suunnitteluvaiheessa pohdittiin valmiiden sisällönhallintajärjestelmien käyttöä, mutta lopulta päädyttiin luomaan oma hallintasivusto, joka sopii kauppapaikan vaatimuksiin. Yksi syy valmiiden sisällönhallintajärjestelmien hylkäämiseen oli niiden päivittämistarve: viime vuosina suuri osa verkkosivustoihin kohdistuneista hyökkäyksistä tapahtui juuri sisällönhallintajärjestelmien kautta ja ne siksi vaativat jatkuvaa päivittämistä uusimpaan, korjattuun versioon.

Tämä opinnäytetyö käsittää teknisen toteutuksen Mikko Jämbäckin tekemään visuaaliseen toteutukseen.

2 SÄHKÖISEN KAUPPAPAIKAN VAATIMUKSET

Sähköisen kauppapaikan vaatimukset selvitettiin toimeksiantajan kanssa ensimmäisenä, ennen mitään muuta suunnittelua. Vaatimusten pohjalta tehtiin vaatimusmäärittely, jossa eriteltiin kolmella eri tasolla toiminnalliset ja ei-toiminnalliset vaatimukset. Nämä kolme tasoa ovat

- Välttämättömät vaatimukset
- Hyödylliset vaatimukset
- Mahdolliset vaatimukset

Tässä toteutuksessa toteutettiin kaikki välttämättömät vaatimukset ja osa hyödyllisistä vaatimuksista. Työn edetessä vähiten tärkeään luokkaan, mahdollisiin vaatimuksiin, syntyi lisää vaatimuksia, jotka käsittivät pääasiassa hallintajärjestelmän helppokäyttöisyysominaisuuksia.

Välttämättömiin vaatimuksiin sisältyivät sellaiset sivuston ominaisuudet, joita ilman sivusto ei toimi tarkoitetulla tavalla. Tällaisia ovat muun muassa tuotteen kuuluminen johonkin tuoteryhmään, tuoteryhmän kuuluminen johonkin tuotesivuun ja tuotteen ominaisuudet. Kaikki tietoturvaan liittyvät vaatimukset kuuluvat tähän luokkaan.

Hyödyllisiin vaatimuksiin kuuluu pääasiassa hallintajärjestelmän ominaisuuksia, kuten ylläpitäjän lataaman tuotekuvan pienentäminen ja vesileiman lisääminen kuvaan, tuotesivuilla olevien tekstien muokkaaminen ja niin edelleen. Tähän luokkaan kuuluvat ominaisuudet toteutettiin vasta, kun välttämättömät vaatimukset olivat täyttyneet.

Mahdollisiin vaatimuksiin sisältyy sellaiset ominaisuudet, jotka eivät ole millään tavalla välttämättömiä sivuston toiminnan kannalta ja jotka mahdollisesti toteutetaan myöhemmässä vaiheessa. Esimerkkinä tähän luokkaan kuuluvasta ominaisuudesta olkoon tuotteiden lajittelu.

Näiden vaatimusten pohjalta laadittiin erilliset vaatimukset palvelimelle, jonka perusteella toimeksiantaja vertaili palveluntarjoajia ja valitsi niistä sopivimman.

2.1 Palvelinalusta

Palvelin päädyttiin toteuttamaan LAMP-alustalle, joka sisältää GNU/Linux-käyttöjärjestelmän, Apache-HTTP-palvelimen, MySQL-relaatiotietokannan ja PHP-komentosarjatulkin. Alustan kaikki ohjelmistot ovat avoimen lähdekoodin projektien aikaansaannoksia, maksuttomia ja niitä päivitetään säännöllisesti. Ohjelmistot ovat suosittuja ja suurin osa Internetin palveluista käyttää niitä. Ohjelmistojen valintaan päädyttiin niiden helpon asennettavuuden ja hyvän dokumentoinnin vuoksi.

Sivuston kehityksen aikana käytettiin testipalvelinta, jossa edellämainitut ohjelmistot oli asennettu Ubuntu Server –käyttöjärjestelmään. Sattumalta asennetut versiot olivat samat kuin palveluntarjoajan Cloud Linux –palvelimella. PHP-koodi luotiin GNU/Linuxin Nano-tekstieditorilla suoraan palvelimella, jolloin koodiin tehdyt muutokset näkyivät heti, eikä tiedostoja tarvinnut siirtää jokaisen muutoksen jälkeen työasemalta palvelimelle. Tekstieditorissa on myös mahdollisuus värikoodata ohjelmakoodi, joka helpottaa ohjelmarakenteen hahmottamista ja virheiden etsimistä.

2.1.1 Apache

Apache on vanhimpia edelleen käytössä olevia web-palvelinohjelmistoja ja sitä kehitetään ja ylläpidetään edelleen. Apache on Netcraftin tutkimuksen mukaan käytetyin palvelinohjelmisto noin 60 % osuudella. Ohjelmisto sisältää runsaasti ominaisuuksia joita kaikkia ei tarvita kauppapaikan toteutukseen. Web-palvelinohjelmisto on se osa toteutusta, joka vastaanottaa HTTP-pyyntöjä asiakkaan selaimelta, suorittaa tarvittavat komentosarjat PHP-moduulin avulla ja lähettää sitten vastaukset selaimelle. Jotkut ovat kritisoineet Apachen tapaa käsitellä HTTP-pyyntöjä vanhanaikaiseksi ja resursseja tuhlaavaksi, mutta nämä ongelmat tulevat esiin vasta, kun käyttäjiä on tuhansia.

2.1.2 PHP

PHP Hypertext Preprocessor (PHP) on alun perin grönlantilaisen Rasmus Lerdorfin vuonna 1995 kehittämä, mutta myöhemmin avoimen kehittäjäjoukon ylläpitämä ohjelmointikieli. PHP muistuttaa paljon Perliä, mutta sen käyttö rajoittuu dynaamisten web-sivujen luomiseen. PHP on enemmän skriptikieli kuin varsinainen ohjelmointikieli, sillä sitä ei tarvitse kääntää konekielelle ennen suorittamista, kuten esimerkiksi C++. Suuremmissa palvelimissa voidaan käyttää valmiiksi konekielelle käännettyjä PHP-skriptejä. Facebook on jopa luonut kokonaan oman kääntäjän, jonka kerrotaan vähentävän palvelimen kuormitusta jopa 50 %. Tässä työssä ei kuitenkaan nähty tarpeelliseksi käyttää kääntäjää.

Aluksi tehty koodi ei alkanut toimia, vaan antoi virheilmoituksen ensimmäisellä rivillä, riippumatta skriptin sisällöstä. Pian kävikin ilmi, että virheet johtuivat Windowsin tavasta käyttää rivinvaihtoon kahden tavun pituista CR LF (Carriage return, Line feed) -ohjausmerkintää, kun Linux käyttää rivinvaihtoon yhden tavun

pituista LF-ohjausmerkkiä. Kauppapaikan ohjelmoinnissa käytettiin tulostuksen puskurointia, joka joissakin tapauksissa parantaa suorituskykyä merkittävästi ja joka mahdollista joustavamman otsikkotietojen, kuten esimerkiksi evästeiden, hallinnan. PHP tallentaa ohjelmiston antaman tulosteen puskurimuistiin, joka lähetetään käyttäjän selaimelle sen tultua täyteen tai ohjelmakoodin kutsuessa puskurin tyhjennystä. Tällainen kutsu annetaan ohjelmakoodin lopussa ja joissain poikkeustapauksissa, kuten pääkäyttäjän uloskirjautumisessa.

Palvelimella käytössä oleva PHP-versio 5.3 on viimeisiä, joka sisältää vanhentuvan mysql-kirjaston. Uudempi mysqli-kirjasto sisältyy myös tähän ja aikaisempiin versioihin ja tulevaisuutta ajatellen kauppapaikan PHP-koodissa käytettiin tätä uudempaa kirjastoa. Mysqli-kirjasto tukee MySQL-palvelimen versiosta 4.13 lähtien käytössä olleita toimintoja, jotka eivät sisälly vanhempaan mysql-kirjastoon. Tuki PHP:n 5.3-versiolle päättyy heinäkuussa 2014, joten palveluntarjoaja todennäköisesti päivittää PHP:n uudempaan versioon siihen mennessä.

2.1.3 MySQL

MySQL on suomalaisen Michael "Monty" Wideniuksen ja ruotsalaisen David Axmarkin vuonna 1996 julkaisema ja nykyään Oraclen kehittämä SQL-relaatiotietokantapalvelin. Palvelimen muiden osien ohella myös MySQL on erittäin suosittu ohjelmisto dynaamisten palvelinohjelmistojen joukossa. MySQL-tietokantaa päädyttiin käyttämään sen maksuttomuuden ja hyvän dokumentaation ja suorituskyvyn vuoksi. Palvelimesta on olemassa myös maksullinen versio, joka sisältää lisäominaisuuksia, mutta sen käyttöä ei nähty tarpeelliseksi.

2.2 Tietoturva

Tietoturva on yksi ohjelmisto- ja web-alan aliarvostetuista asioista. Ilman riittävää tietoturvaa ulkopuoliset saattavat saada pääsyn yrityksen kriittisiinkin tietoihin, kuten asiakasrekistereihin. Tietoturvan parantaminen ei kuitenkaan ole kovinkaan vaikeaa ja PHP:n ja MySQL:n tapauksissa se onnistuu muutamalla yksinkertaisella tavalla. Sähköinen kauppapaikka poikkeaa tavanomaisesta verkkokaupasta siten, että tuotteet on listattu web-sivulla, mutta koska niitä ei suoraan tilata webistä, PHP-skripteille ei tarvitse antaa kirjoitusoikeutta tietokantaan.

2.2.1 SQL-tietoturva

SQL-tietokantojen yleisin tietoturvaongelma on SQL-injektio. SQL-injektiossa pyynnön tekevä ohjelmakoodi, tässä tapauksessa PHP, ei tarkista riittävän huolellisesti käyttäjän antamaa syötettä. Vihamielinen käyttäjä, yleensä hakkeri, lisää syötteeseen haitallista SQL-koodia, jolla hän pyrkii suorittamaan ylimääräistä SQL-koodia ja saamaan haltuunsa tai poistamaan tietueita. Tutkimuksen mukaan toiminnassa olevat sivustot kokevat keskimäärin 71 SQL-injektioyritystä tunnissa ja ne ovat selvästi yleisin (83 %) onnistuneiden hakkerointiyritysten syy (Imperva 2011).

Aiemmin mainittiin, ettei tavallinen käyttäjä tarvitse kirjoitusoikeutta tietokantaan. Tämä parantaa huomattavasti tietoturvaa, sillä vaikka vihamielinen käyttäjä onnistuisikin ohittamaan syötteentarkistuksen, hän pystyisi vain lukemaan tietueita. Kirjoitusoikeudet tietokantaan ovat ainoastaan pääkäyttäjällä. Näin hän pystyy muokkaamaan tuotteiden tietoja ja tarvittaessa vaihtamaan salasanansa. Tietokannan käyttäjätaulun salasanat ovat vahvasti salattuja:

salasanoista muodostetaan tiiviste modernilla SHA-256 –algoritmilla käyttämällä pisintä algoritmin sallimaa suolamerkkijonoa, jonka pituus on 16 merkkiä. Salasanan ja suolan yhdistelmä silmukoidaan 5000 kertaa algoritmin läpi, jolloin muodostettua tiivistettä on käytännössä mahdotonta murtaa. Koska algoritmi on yksisuuntainen, sillä voidaan ainoastaan luoda tiivisteitä, muttei avata niitä.

Pääkäyttäjän kirjautuessa annetusta salasanasta muodostetaan samalla tavalla tiiviste, jota sitten verrataan tietokantaan tallennettuun tiivisteeseen. Salasanan ollessa väärä ohjelmisto viivyttää syötteen tulostusta kaksi sekuntia, joka hidastaa huomattavasti murtautumisyrityksiä, mutta joka käyttäjän kannalta ei ole liian häiritsevä. Yleinen ohjelmointivirhe, joka salasanan tarkistuksessa syntyy, on antaa salasana sellaisenaan ilman tarkistuksia SQL-palvelimelle, jolloin hyökkääjä voi sisällyttää salasanaan SQL-lauseen, joka palauttaa tosi-arvon. PHP-koodi päättelee tällöin virheellisesti, että salasana oli oikein ja antaa hyökkääjälle vapaan pääsyn tietoihin.

2.2.2 PHP-Tietoturva

Olennaisin osa PHP:n tietoturvaa on käyttäjän syöttämien tietojen oikeellisuustarkistus. Käyttäjän tai sen selaimen antamiin syötteisiin ei pidä koskaan luottaa, vaan tarkistaa ne mahdollisten SQL-injektioiden varalta. Oikeellisuustarkistuksista yleisin on sivuilla käytettyjen URI-parametrien tarkistus: näin estetään tehokkaasti SQL-injektiot. Selaimen antamista parametreista id-numerot tarkistetaan Ctype-kirjaston ctype_digit-funktiolla, joka tarkistaa, onko annettu parametri kokonaislukumuotoa. Kaikki tietokantahakuihin liittyvät syötteet ajetaan SQL-injektioiden varalta Mysqli-kirjaston real_escape_string-funktiolla, joka poistaa erikoismerkkien, kuten lainausmerkkien ja puolipilkkujen, erityismerkityksen (Meloni 2012, 362–365).

Sivuston yleisessä osiossa lomaketietoja käytetään ainoastaan yhteydenottolomakkeessa. Voidakseen lähettää lomakkeen tiedot käyttäjän tulee antaa vastaus yksinkertaiseen kysymykseen. Tällaisia kysymyksiä ovat esimerkiksi ”Minkä värinen on taivas iltapäivällä” tai ”Milloin Kuu näkyy parhaiten”. Käyttäjän lomakkeeseen syöttämät tiedot tarkistetaan ja ajetaan strip_tags-funktion läpi, joka poistaa HTML-koodin syötteen seasta, ja vasta sitten annetaan ne sähköpostijärjestelmän välitettäväksi (Meloni 2012, 174). HTML-koodin poistaminen on tärkeää siksi, ettei vihamielinen käyttäjä pysty upottamaan syötteen sekaan omaa HTML-koodia, joka voisi esimerkiksi sähköpostia avatessa ladata kolmannen osapuolen JavaScript-koodia, joka puolestaan voi ladata häiritsevää sisältöä ja pahimmassa tapauksessa haittaohjelmia. (The PHP Group. 1997–2014.)

Hallintasivulla on käytössä samantasoinen tietoturva kuin muuallakin sivustolla. Tämän lisäksi käyttäjän todennus on luonnollisesti vahvempi: kirjautumisen jälkeen käyttäjän selaimeen asetetaan eväste, jonka avulla käyttäjä tunnistetaan samaksi HTTP-pyyntöjen yhteydessä (Meloni 2012, 213–214). PHP sisältää istunnonhallintakirjaston, joka käsittää istuntoon liittyvien tietojen, kuten edellisen vierailukerran tallentamisen, ja samalla hoitaa evästeiden asettamisen. Pääkäyttäjä voi asetuksissaan määrittää käyttämättämänäolon aikakatkaisuarvon, jonka kuluttua umpeen pääkäyttäjä joutuu kirjautumaan uudelleen. SSL-suojatun HTTPS-yhteyden käyttöä hallintasivustolla ei nähty tarpeelliseksi, mutta se voidaan kohtuullisella vaivalla ottaa käyttöön myöhemmin.

3 SIVUSTON RAKENNE

Sivuston valikkorakenne on staattinen. Useimmat sivut ovat pääkäyttäjän muokattavissa kulloisenkin tarpeen mukaan. Hallintasivut sisältävät tarvittavat työkalut sisällön muokkaamiseen. Työssä päädyttiin laatimaan oma hallintasivusto sisällön hallitsemiseksi. Saatavilla olevat sisällönhallintajärjestelmät olisivat myös soveltuneet sisällön muokkaamiseen, mutta niiden käyttöönotto ja ylläpitäminen olisivat olleet hankalampia kuin kustomoidun hallintasivuston laatiminen. Tietoturvan kannalta tulee muistaa, että ohjelmointivirheiden määrä on suoraan verrannollinen ohjelmakoodin määrään nähden: myös siksi yksinkertaistettu hallintasivusto toimii esimerkissämme paremmin kuin valmis sisällönhallintajärjestelmä.

Sivuston tuotesivut sisältävät suurimman osan dynaamisesta sisällöstä. Pääkäyttäjä pystyy helposti lisäämään ja muokkaamaan tuotteita hallintasivustolla. Tuotteiden tiedot kuvatiedostoja lukuun ottamatta sijaitsevat tietokannassa, josta PHP-skripti hakee ne jokaisen sivulatauksen yhteydessä. Myös muiden dynaamisesti luotujen sivujen tiedot ladataan tietokannasta. Ohjelma ei lajittele tuotesivulle haettavia tuotteita, vaan MySQL-palvelin antaa ne lisäysjärjestyksessä. Tuotteiden lajittelu voidaan myöhemmin lisätä helposti lisäämällä SQL-komentoon lajitteleva SORT BY –lause. Asiakas voi näin lajitella tuotteet esimerkiksi aakkos- tai hintajärjestykseen.

HTML5-standardi on tuomassa mukanaan uusia elementtejä, jotka helpottavat tietojen syöttämistä. Tällaisia lomake-elementtejä ovat muun muassa sähköposti-, puhelin- ja erilaiset numeeriset elementit (McDaniel 2012, 126–128). Joitakin näistä elementeistä hyödynnetään hallintasivustossa helpottamaan tuotteiden ja asetusten muokkaamista. Vuoden 2014 loppupuolella julkaistava standardi

poistaa käytöstä joitakin HTML-muotoiluja ja siirtää niiden käytön Cascading Style Sheet (CSS)-tyylisivun hoidettavaksi. Käytöstä poistuvia muotoiluja ei ole montaa eivätkä ne vaikuta kauppapaikan koodiin. Yksi HTML5-standardin mielenkiintoisista uutuuksista on tuki Drag and Drop- eli vedä ja pudota–toiminnolle (McDaniel 2012, 234). Toistaiseksi se ei ole käytössä kauppapaikassa, mutta sen avulla sivuston hallintaa voitaisiin helpottaa.

3.1 Ohjelmakoodin rakenne

Sivuston ohjelmointi toteutettiin proseduraalisella ohjelmointitavalla pääasiassa tottumuksen vuoksi. PHP tukee täysin myös oliopohjaista ohjelmointitapaa ja joissakin tilanteissa se onkin suositeltavampi tapa. Suorituskyvyn kannalta ohjelmointitavalla ei ole ainakaan huomattavaa merkitystä.

Sivuston ohjelmakoodi koostuu sisäkkäisistä if-ehtolauseista. Aivan ensimmäisenä, ennen varsinaisen ohjelmakoodin alkua, aloitetaan syötteen puskurointi. Puskuroinnin avulla PHP-skriptin suoritus ja käyttäjälle lähetettävän tiedon lähettäminen nopeutuu. Nopeusero on merkittävä erityisesti runsaasti tietokantakyselyjä sisältävissä skripteissä. HTTP-vastauksen alussa ennen varsinaista sisältöä palvelin lähettää otsikkotietoja, joita ei voida lähettää, jos sisältöä on jo lähetetty käyttäjän selaimelle. Puskurointi tarjoaa mahdollisuuden muuttaa otsikkotietojen sisältöä, kuten esimerkiksi evästeitä, vaikka ohjelmakoodi olisikin jo tulostanut sisältöä. Tämä on erityisen hyödyllinen ominaisuus istuntojen käsittelyssä.

Ensimmäisessä varsinaisessa ehtolauseessa ohjelmisto yhdistää tietokantapalvelimeen ja antaa virheilmoituksen, jos yhdistäminen epäonnistuu (Meloni 2012, 358–359). Tietoturvasyistä tarkkoja tietoja epäonnistumisen syystä ei

pidä antaa, sen sijaan ohjelma vain ilmoittaa tietokantayhteysvirheen tapahtuneen ja kehottaa käyttäjää yrittämään hetken kuluttua uudestaan. Seuraavana asetetaan käytettävä merkistö UTF-8 –merkistöön, jotta vältyttäisiin erilaisten merkistöjen sekakäytöstä johtuvista ongelmista. Selaimelle lähetetään otsikkotiedoissa tieto käytetystä merkistöstä ja tietokantayhteys käsketään käyttämään UTF-8–merkistöä mysqli_set_charset-funktiolla.

Alkutoimenpiteitä seuraavassa ehtolauseessa tutkitaan, lähettikö asiakkaan web-selain pyynnön GET- vai POST-metodilla. GET-pyynnöllä käyttäjän selain pyytää palvelinta lähettämään tietyn tiedoston ja POST-pyynnöllä selain lähettää palvelimelle käyttäjän syöttämiä tietoja. Kummassakin tapauksessa ohjelmisto selvittää asiakkaan pyytämän alasivun switch-case –silmukan avulla. Tämän jälkeen ohjelmisto suorittaa pyydetyn alasivun ohjelmakoodin. Mikäli alasivua ei voida selvittää tai sitä ei pyydetty, ohjelma tulostaa sivuston etusivun.

If-ehtolauseita suositellaan käyttämään ohjelmakoodin perusrakenteena joissakin yhteyksissä, kuten esimerkiksi tietokantayhteyden muodostamisessa tai muissa tilanteissa, joissa jonkin funktion tuloksena voi olla epätosi (Meloni 2012, 359). Virheen sattuessa voidaan tulostaa virheilmoitus suorituksen siirryttyä ehtolauseen else-osaan. Ohjelmakoodin suoritus ei siis koskaan päädy umpikujaan tai pääty kesken eikä se näin ollen tuota yllättäviä poikkeuksia.

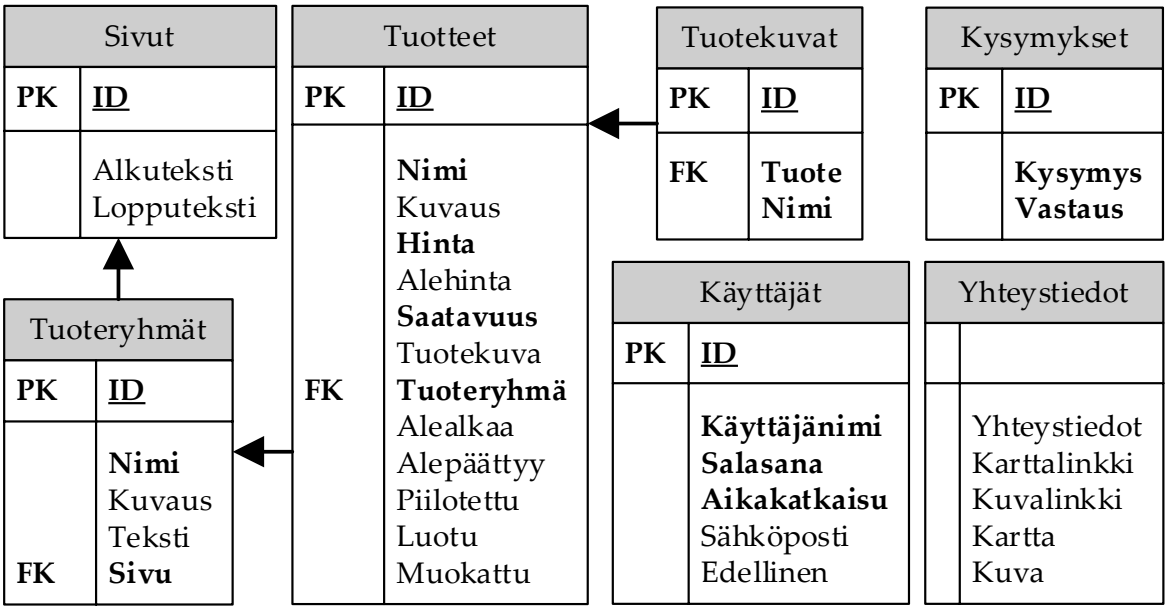
3.2 Tietokannan rakenne

Aiemmin laaditun vaatimusmäärittelyn pohjalta laadittiin tietokantamäärittely. Sen tekemiseksi pohdittiin, millainen rakenne sivustolla ja sen alasivuilla on. Pohdinnassa tultiin seuraaviin johtopäätöksiin:

- Alasivuilla on ryhmiä
- Alasivuilla on lyhyt johdanto
- Ryhmillä on myös lyhyt johdanto
- Tuotteet kuuluvat johonkin ryhmään
- Tuotteilla on niitä kuvaavia tietoja
- Tuotteilla on kuvia

Näiden tietojen pohjalta laadittiin kaavio kaikista tietokantaan sisältyvistä tauluista ja niiden sarakkeista sekä niiden riippuvuuksista. Tietokantaa ei erityisemmin normalisoitu, mutta koska tietokannan riippuvuudet ovat pieniä ja yksi-moneen -riippuvuuksia, on se valmiiksi normalisoitu. Tietokanta sisältää lisäksi hallintasivuston käyttäjätunnukset ja niiden tiedot.

Kuviossa 1 on kuvattu tietokannan taulut sarakkeineen. Yhteystieto-taulua lukuun ottamatta kunkin taulun ensimmäinen rivi koostuu avaimesta (Primary Key, PK) (Meloni 2012, 302). Lihavoidut kentät ovat pakollisia sarakkeita, joita ei voi jättää tyhjäksi. Poikkeuksena tähän ovat ID-sarakkeet, joiden arvo luodaan automaattisesti tietuetta lisättäessä: taulua luotaessa ID-sarakkeen määrityksessä on annettu AUTO_INCREMENT-määrite. Taulujen väliset viittaukset (relaatiot) on esitetty taulusta toiseen osoittavilla nuolilla: nuolen alkukohdassa olevan taulun vierasavain (Foreign Key, FK) viittaa nuolen kohteena olevan taulun primääriavaimeen (Primary Key, PK). Esimerkkinä tällaisesta viittauksesta olkoon Tuotteet-taulussa sijaitseva Tuoteryhmä-sarake, jonka arvo on sama kuin tuoteryhmän, johon tuote kuuluu, ID-sarakkeen arvo.



KUVIO 1. Tietokannan rakenne ja relaatiot

3.3 Hallintasivusto

Hallintasivusto koostuu pääosin sivustonhallintajärjestelmästä ja sen ulkoasu rakennettiin noudattamaan muun sivuston ulkoasua. Sivuston vasemman reunan valikko korvattiin hallintasivuston omalla valikolla. Valikko sisältää seitsemän tärkeintä kohtaa, joista osa sisältää alakohtia. Alakohtia ei näytetä valikossa, vaan valikosta avautuvilla sivuilla. Alakohdat koottiin aihepiireittäin kullekin sivulle.

Hallintasivustolla ylläpitäjä voi lisätä, muokata ja poistaa tuotteita ja tuoteryhmiä, hallita käyttäjätiliänsä ja muokata sivustolla näytettäviä tietoja, kuten yhteystietoja. Lisäksi on mahdollista vaihtaa sivuston otsikkokuva ja piilottaa tuotteita esimerkiksi muokkauksen tai saatavuusongelmien vuoksi.

Kaikissa hallintasivuston lomakkeissa käytetään POST-metodia, jolloin lomakkeen tiedot lähetetään HTTP-pyynnön rungossa. Toisin kuin GET-metodilla,

lomakkeen kenttien parametrit eivät sisälly URI-osoitteeseen, jolloin sivun uudelleenlataus ei toista esimerkiksi tuotteen tai tuotekuvan poistoa. GET-metodia käytetään muissa yhteyksissä, kuten esimerkiksi alisivujen latauksissa. GET-metodissa palvelimelle välitettävät parametrit sisältyvät URI-osoitteeseen ja ohjelmakoodi tulostaa näiden parametrien mukaan sisällön.

3.3.1 Kirjautuminen hallintosivustolle

Pääkäyttäjän kirjauduttua hallintasivustolle ohjelmakoodi käynnistää istunnon PHP:n istunnonhallintakirjastoa käyttäen (Meloni 2012, 218). Palvelimelle tallennetaan pieni evästettä muistuttava tekstitiedosto, joka sisältää tietoja istunnosta ja istuntoon liitettyjä muuttujia (Meloni 2012, 219–222). Tällaisia muuttujia on muun muassa aikaleima, jolloin istunto vanhenee. Käyttäjän ollessa aktiivinen palvelin tarkistaa, onko aikakatkaisun aikaleima vanhentunut ja sen perusteella tallentaa muuttujaan uuden aikaleiman tai näyttää käyttäjälle kirjautumissivun. Käyttäjän kirjautuessa ulos istunto tietoineen tuhotaan (Meloni 2012, 223–224).

Kirjauduttaessa käyttäjälle näytetään edellisellä kirjautumiskerralla tallennettu kirjautumisaika ja mikäli käyttäjätilin asetuksissa on niin määritetty, myös IP-osoite. Käyttäjän selaimeen tallennetaan eväste, joka sisältää istunnon 32 merkkiä pitkän satunnaisen heksadesimaalimuotoisen tunnisteen. Eväste on pieni selaimeen tallennettava tekstitiedosto, jonka perusteella käyttäjä tunnistetaan samaksi sivujen latauksen välillä. Valtaosa web-sivuista käyttää evästeitä erilaisiin käyttötarkoituksiin. Eväste voi sisältää kirjautuneen käyttäjän tietoja, kuten tässä tapauksessa, tai tietoja esimerkiksi käyttäjän vierailemista sivustoista. Evästeet ovat palvelin- tai domainkohtaisia ja se lähetetään palvelimelle jokaisen sivun tai tiedoston latauksen yhteydessä. Evästeet ovat haitattomia, sillä ne eivät sisällä

suoritettavaa ohjelmakoodia eikä niitä hyödyntäviä haittaohjelmia tunneta. Sen sijaan erilaiset mainospalvelut voivat käyttää niitä käyttäjän selailutottumusten seuraamiseen ja mainonnan kohdentamiseen. (Meloni 2012, 213–214.)

3.3.2 Tuotesivu ja -ryhmät

Sivustolla on kolme tuotesivua erilaisia tuotteita varten. Kullakin sivulla on tuoteryhmiä, joihin voidaan lajitella tuotteita erilaisten ominaisuuksien tai käyttötarkoitusten perusteella. Teknisesti on mahdollista luoda useampiakin tuotesivuja: liian suuri määrä kuitenkin monimutkaistaa sivustoa ja tekee tuotteiden löytämisestä hankalaa. Tuoteryhmiä sen sijaan voidaan lisätä rajattomasti.

Sivuja ja ryhmiä voidaan muokata ja poistaa aivan kuten tuotteitakin. Koska jokaisen ryhmän tulee kuulua johonkin sivuun ja jokaisen tuotteen johonkin ryhmään, voisi sivujen tai ryhmien poistaminen saada aikaan orpoja ryhmiä ja tuotteita, tulee sivujen ja ryhmien poistaminen olla mahdollista ainoastaan silloin, kun niissä ei ole ryhmiä tai tuotteita. Siksi tuotesivuja tai –ryhmiä poistettaessa varmistetaan, ettei orpoja ryhmiä tai tuotteita pääse syntymään ja tarvittaessa käyttäjältä kysytään vahvistusta. Vaihtoehdoiksi annetaan ryhmään kuuluvien tuotteiden poistaminen kuvineen, siirtäminen toiseen tuoteryhmään tai toiminnon peruuttaminen.

3.3.3 Tuotteiden muokkaaminen ja tuotekuvien lisääminen

Tuotesivujen ja –ryhmien muokkaamisen lisäksi ylläpitäjä voi lisätä ja muokata tuotteita. Luonnollisesti tuotteella tulee olla nimi, hinta ja saatavuusaika.

Tuotteelle annetaan myös tarvittaessa lyhyt kuvaus. Tuote voidaan asettaa myös alennukseen, joka on voimassa ennalta määritellyn ajan. Järjestelmä näyttää asiakkaille alennetun hinnan alennuksen ollessa voimassa ja muulloin näytetään normaalihinta. Tuotteet voidaan piilottaa esimerkiksi tuotteen muokkaamisen ajaksi, jolloin asiakkaat eivät näe tuotetta ilman tuotekuvia tai epämääräisin kuvauksin tai hinnoin. Tuotteita voi myös poistaa, jolloin kaikki tuotteen tiedot poistetaan tietokannasta ja tuotekuvat tuhotaan.

Kullakin tuotteella voi olla enintään viisi tuotekuvaa, joista yksi toimii tuotelistauksessa näytettävänä oletuskuvana. Loput kuvat näytetään tuotteen tietosivulla saatavuus- ja muiden tietojen ohessa. Ohjelmisto pienentää ladatun kuvan automaattisesti sopivaan kokoon, jonka päätettiin olevaan 800 × 600 pikseliä. Ylläpitäjä voi kuvaa ladatessa valita, lisätäänkö kuvaan vesileima. Vesileima on kuvassa näkyvä vaalea pieni kuva, jonka tarkoitus on estää tai ainakin hankaloittaa kuvien luvatonta käyttöä, mutta olla kuitenkin mahdollisimman vähän tavallista käyttöä häiritsevä. Lopuksi kuvasta tehdään pienempi versio ilman vesileimaa tuotelistausta varten.

Tuotekuva lisätään valitun tuotteen tietosivulta. Käyttäjä valitsee haluamansa kuvan, joka sitten ladataan palvelimelle. Nykyaikaisilla digitaalikameroilla voi ottaa hyvin tarkkoja kuvia, joka luonnollisesti kasvattaa kuvatiedoston kokoa. Joskus liian suuri tiedosto voi aiheuttaa hankaluuksia ladatessa sitä palvelimelle: palvelimen konfiguraatiossa on useampi ladattavia tiedostoja koskeva asetus ja ymmärrettävästi pienin niistä määrittää suurimman sallitun tiedostokoon.

Jotta pääkäyttäjä voisi ladata tiedostoja palvelimelle, täytyy lomakkeen metodiksi määrittää POST ja enkoodaustyypiksi multipart/form-data (Meloni 2012, 207). Tämä on ainoa tyyppi, jonka avulla voidaan lähettää binäärimuotoista dataa. Mikäli enkoodaustyyppiä ei ole määritetty, web-selaimet lähettävät lomakkeen

pelkästään tekstimuotoista dataa välittävää application/x-form-urlencoded–muotoa käyttäen, jolloin binäärimuodossa esitettävä data, esimerkiksi kuvat, eivät välity palvelimelle.

Vastaanottaessaan pääkäyttäjän lähettämän lomakkeen, PHP tarkistaa aivan ensimmäisenä onko lomakkeessa suurimman tiedostokoon määrittävää MAX_FILE_SIZE-kenttää. Tämän kentän arvoon ei kuitenkaan voida sokeasti luottaa, sillä käyttäjä voi vaihtaa muuttujan arvon mieleisekseen. Tämän kentän tarkoituksena on keskeyttää lomakkeen vastaanotto välittömästi, mikäli sen koko ylittää kenttään syötetyn arvon.

Lähetetyn lomakkeen sisältäessä tiedostoja PHP luo globaalin, kaikkialta suoritettavasta ohjelmakoodista tavoitettavan $_FILES-taulukon, joka sisältää tietoja ladatusta tiedostosta. Tällaisia tietoja ovat tiedoston alkuperäinen nimi, MIME-tyyppi, koko, väliaikainen tiedostonimi ja virhekoodi. Taulukko on kaksiulotteinen, joten yhdellä lomakkeella voitaisiin lähettää useampi tiedosto samalla kertaa. Taulukon ensimmäisen sarakkeen nimi määräytyy lomakkeessa olleiden tiedostokenttien nimien mukaan.

Palvelimen konfiguraatiossa on määritetty suurin sallittu ladattava tiedostokoko upload_max_filesize-muuttujassa. Max_execution_time-muuttujassa määritetty enimmäissuoritusaika rajoittaa myös tiedostokokoa etenkin hitailla yhteyksillä. Enimmäisajan täyttyessä lataus keskeytyy ja ohjelmakoodin suoritus jatkuu. Tiedostojen latauksen keskeytyessä sallitun tiedostokoon, muun rajan täyttyessä tai virheen tapahtuessa PHP sijoittaa virhekoodin $_FILES-taulukon error-soluun. Onnistuneen latauksen virhekoodi on 0 ja sen ollessa jokin muu voidaan päätellä, mikä virhe on tapahtunut ja antaa pääkäyttäjälle tilanteeseen sopiva virheilmoitus ja ehdottaa keinoja virheen välttämiseksi.

HTML5-standardin edistymispalkki voidaan ottaa käyttöön kuvien latauksissa palveluntarjoajan päivittäessä PHP-ohjelmiston uudempaan versioon (McDaniel 2012, 50–51). Nykyinen ohjelmistoversio keskeyttää ohjelmakoodin suorituksen tiedostojen latauksen ajaksi, jolloin asiakkaalle ei voida lähettää tietoa latauksen edistymisestä, eivätkä selaimet sisällä toimintoja latauksen edistymisen tarkkailuun. Tämä toteutetaan JavaScriptin XmlHttpRequest-metodilla, joka hakee lyhyin välein palvelimelta käynnissä olevan latauksen tiedot ja muotoilee vastaanotetun datan käyttäjälle esitettävään muotoon (Negrino & Smith 2007, 363–368).

Tiedostomuodoista käytettäviksi valittiin yleisimmin käytetyt ja nykyaikaisimmat JPEG- ja PNG-muodot. Vanhentunutta ja rajoittunutta GIF-muotoa sekä pakkaamatonta BMP-muotoa ei sivustolla käytetä. Tiedoston latauduttua palvelimelle hallintasivuston skripti tarkistaa ladatun kuvan tyypin Exif-kirjaston exif_imagetype-metodilla. Muita kuin JPEG- ja PNG-kuvia ladattaessa kuva hylätään ja käyttäjälle annetaan ilmoitus väärästä tiedostomuodosta.

3.3.4 Tuotekuvien vesileima

Vesileiman toteutus on teoriassa yksinkertaista. PHP sisältää funktiokirjastoja kuvien muokkaamiseen ja yksi yleisimmin käytetyistä kuvankäsittelykirjastoista on GD. GD-kirjaston imagecopymerge-funktiolla voidaan kaksi kuvaa liittää päällekkäin ja funktion pct-parametrilla voidaan määrittää kuvien läpinäkyvyys asteikolla 0-100. Käyttämällä sopivaa arvoa pct-parametrille, saadaan toinen kuva liitettyä osittain läpinäkyvänä, jolloin alkuperäiseen kuvaan muodostuu vesileima.

Toteutuksessa käytetään kahta erikokoista vesileimakuvaa. Pienempää käytetään, jos pääkäyttäjän lataama kuva ei ole riittävän suuri isoa vesileimaa varten ja

isompaa kuvaa, jos ladattu kuva on riittävän suuri. Iso vesileima lisätään kuvaan ennen lopullista pienentämistä, jolloin vältytään sahalaitaisuuksilta lopullisessa kuvassa.

Toteutusvaiheessa havaittiin, ettei vesileimakuva PNG-kuvana läpinäkyvällä taustalla toiminutkaan toivotulla tavalla. Ongelmaa ratkaistaessa huomattiin, ettei imagecopymerge-funktio tuekaan PNG:n sisältämää alfakanavaa. Nimensä, GIF Drawing, mukaisesti päädyttiin ratkaisemaan ongelma käsittelemällä PNG-kuvaa GIF-kuvan tavoin: GIF-kuvassa läpinäkyvyys määritetään joksikin RGB-värin arvosta. Koska vesileimassa esiintyy ainoastaan vaaleita värisävyjä, läpinäkyväksi väriksi määritettiin imagecolortransparent-funktiolla musta (R:0,G:0,B:0). Pian huomattiin ratkaisun toimivan. Myöhemmin voidaan siirtyä käyttämään nykyaikaisempaa Cairo-kirjastoa, mikäli esimerkiksi tuki GD-kirjastolle päättyy tulevaisuudessa.

3.4 Asiakassivusto

Asiakassivusto koostuu etusivun lisäksi kolmesta tuotesivusta ja yhteystietosivusta. Asiakas voi selata tuotesivuilla olevia tuoteryhmiä ja tuotteita. Koska kauppapaikka ei ole täysiverinen verkkokauppa, sen kautta ei voi tilata tuotteita. Yhdeksi jatkokehitysajatukseksi nousi esille mahdollisuus merkitä tuotteita suosikeiksi ja lisätä ne automaattisesti yhteydenottopyyntöön kiinnostavina tuotteina, jotka asiakas mahdollisesti haluaa ostaa.

Tuotesivuilla näytetään kunkin tuotesivun sisältämät tuoteryhmät ja niihin sisältyvät tuotteet. Tuotteista näytetään nimen ja hinnan lisäksi oletuskuva, jota klikkaamalla oletuskuva avautuu suurennettuna. Nimeä klikkaamalla asiakas ohjautuu tuotteen sivulle, missä näytetään yksityiskohtaisempia tietoja tuotteesta.

Oletuskuvan avautumisen aikana XMLHttpRequest-metodi hakee palvelimelta tuotteen loppujen kuvien nimet ja lisää ne tuotteen kuvasarjaan (Negrino ym. 2007, 363–368). Näin menettelemällä voidaan vähentää palvelimen kuormitusta etenkin, jos asiakkaat löytävät sivuston ja se saavuttaa suuren suosion.

Yhteystietosivulla on esillä toiminimen osoite- ja puhelintiedot, joista sähköpostiosoite on sijoitettu ROT13-salausmenetelmällä lähdekoodin sekaan. Sivun latauduttua JavaScript muuntaa sähköpostiosoitteen selkokieliseksi. Yhteystietojen lisäksi sivulle on upotettu Google Maps –kartta, josta asiakas näkee toiminimen sijainnin kartalla ja halutessaan voi myös hankkia ajo-ohjeet toimipisteeseen. Yhteystietosivulta asiakas voi lähettää yhteydenottopyynnön toiminimelle. Asiakas kirjoittaa vapaamuotoisen, enintään 500 merkkiä pitkän viestin ja valitsee, haluaako yhteydenoton sähköpostitse vai puhelimitse. Roskapostirobottien välttämiseksi asiakkaalle esitetään pääkäyttäjän laatima yksinkertainen kysymys, johon oikein vastaamalla lomakkeen sisältö hyväksytään. Asiakas voi halutessaan vaihtaa kysymyksen, jolloin selain noutaa toisen kysymyksen palvelimelta XMLHttpRequest-metodilla ja sijoittaa sen entisen kysymyksen tilalle. Vastausta kysymykseen ei lähetetä selaimelle tarkistettavaksi, vaan tarkistus tapahtuu palvelimella käyttäjän lähetettyä lomakkeen. Lopulta PHP-skripti lähettää lomakkeen tiedot sähköpostitse pääkäyttäjän määrittämään sähköpostiosoitteeseen mail-funktion avulla (Meloni 2012, 202–203).

3.5 JavaScript

Web-sivusta voidaan tehdä dynaaminen JavaScriptin avulla. Sen avulla voidaan muokata web-sivun HTML-koodia Document Object Model DOM:ssa. Sen

suoritus tapahtuu käyttäjän selaimessa ja se mahdollistaa dokumentin elementtien ominaisuuksien ja sisällön muuttamisen ilman, että koko dokumenttia tarvitsisi ladata kokonaan uudelleen. (Negrino ym. 2007, 2.)

DOM yhdessä JavaScriptin kanssa mahdollistaa interaktiivisuuden web-sivuilla (Negrino ym. 2007, 281). Tapahtumien (event) avulla voidaan kutsua JavaScript-koodia, joka toteuttaa halutun toiminnon käyttäjän esimerkiksi siirtäessä kursorin jonkin elementin päälle tai käyttäjän muuttaessa jonkun lomake-elementin sisältöä (Negrino ym. 2007, 219–236).

Kauppapaikka sisältää melko vähän JavaScript-koodia: asiakassivustolla yhteydenottosivulla tarkistetaan, että asiakas on täyttänyt vaadittavat kentät ennen lomakkeen lähettämistä ja että yhteystietokentän otsikko vastaa asiakkaan haluamaa yhteydenottotapaa. Hallintasivustolla JavaScriptiä käytetään enemmän: sen avulla voidaan esittää dialogeja, joilla varmistutaan käyttäjän aikeista poistaa tuotekuvia, tuotteita ja niin edelleen (Negrino ym. 2007, 33–34).

4 TIETOKANTAKYSELYT

MySQL käyttää nimensä mukaisesti standardoitua Structured Query Language (SQL) –kieltä, joka määrittelee tietojen määrittelyn ja käsittelyn tietokannoissa. Joka kerta, kun asiakas tai pääkäyttäjä lataa sivun, PHP-ohjelmakoodi muotoilee tilanteeseen sopivan SQL-kyselyn, jolla se hakee tai muuttaa tietoja tietokannassa. PHP sisältää funktiokirjaston MySQL-tietokantayhteyksille ja niiden avulla onkin helppoa luoda tietokantakyselyjä.

Kun käyttäjä lataa tuotesivun, ohjelmakoodi selvittää ensin tuotesivun sisältämät tuoteryhmät lähettämällä mysqli_query-funktiolla MySQL-palvelimelle komennon `SELECT id, nimi FROM tuoteryhmät WHERE sivu = 1` (Meloni 2012, 304–305). Komento hakee ryhmän id:n ja nimen niistä ryhmistä, jotka kuuluvat sivulle, jonka id-numero on 1. Tietokantapalvelimen mysqli_query-funktiolle lähettämä vastaus tallennetaan muuttujaan, jotta sitä voidaan käyttää. Ennen tietojen käyttöä täytyy selvittää if-lausekkeen avulla palauttiko tietokantapalvelin epätosi-arvon vai tietueita. Jos palvelin palautti epätosi-arvon, ryhmien taulu on joko tyhjä tai komennon syntaksissa on jotain vikaa.

Suorittamalla while-silmukassa mysqli_fetch_assoc-funktiota tietokantakyselyn tulokset noudetaan rivi kerrallaan assosiatiiviseen taulukkoon ja niitä voidaan sitten käyttää tuotteiden noutamiseen kustakin ryhmästä sisäkkäisen while-silmukan avulla. Silmukassa käytetään laskurimuuttujaa, jonka avulla voidaan tulostaa haluttu määrä tuotteita kullekin riville. Näytettävät tuotetiedot voidaan nyt hakea tietokannasta käyttämällä komentoa `SELECT id, nimi, kuva FROM tuotteet WHERE ryhma = $ryhmä['id']`, missä $ryhmä on aiemmin haettu assosiatiivisen taulukko ja ['id'] taulukon id-niminen sarake. Nyt tuotteiden tiedot voidaan tulostaa tuotesivulle. Koska tuotteen oletuskuva on tallennettu suoraan

tuotteen tietoihin, ei kuvien noutamiseksi tarvitse suorittaa erillistä SQL-komentoa jokaisen tuotteen kohdalla.

Hallintasivustolla SQL-komennot ovat osin samoja, kuin itse sivustolla. Näiden lisäksi käytetään myös UPDATE-komentoja tietojen muuttamiseen, INSERT-komentoja tietojen syöttämiseen ja DELETE-komentoja tietojen poistamiseen (Meloni 2012, 316–320). Koska nämä komennot eivät palauta lainkaan tietoja, toisin kuin SELECT-komento, vaan pelkän tiedon kyselyn onnistumisesta, täytyy näiden komentojen onnistuminen tarkistaa tutkimalla, palauttiko tietokantapalvelin true-arvon (Meloni 2012, 361). Muutettujen rivien lukumäärä voidaan noutaa mysqli_afftected_rows-funktiolla. Uusi tuote voidaan lisätä tietokantaan käyttämällä komentoa `INSERT INTO tuotteet SET (nimi, hinta) VALUES ('Uusi tuote', 19.90)`. Tietokantapalvelin antaa uudelle tuotteelle automaattisesti uniikin id-numeron, kuten tietokantaa luodessa taulun määritteisiin on asetettu. Vastaavasti äsken lisätyn tuotteen hintaa voidaan muuttaa komennolla `UPDATE tuotteet SET kuvaus = 'Uuni tuore tuote', hinta = 14.90' WHERE id = 123`.

SQL-komennot ovat melko yksinkertaisia ja suurin vaikeus komentojen käytössä lieneekin niiden sovittaminen kulloisenkin tarpeeseen. Tuotteen tietojen noutaminen on hyvin samankaltainen toimenpide tuoteryhmien noutamisen kanssa, suurin ero lieneekin noudettavien sarakkeiden valinnassa.

5 TULOKSET JA POHDINTA

TT Sisustus tmi:lla ei aiemmin ollut minkäänlaista verkkomarkkinointia. Tekninen toteutus yhdistettynä samaan aikaan toteutettuun Mikko Jämbäckin graafiseen toteutukseen osoittautui toimivaksi kokonaisuudeksi. Työn aikana toimeksiantajaa tavattiin noin kuukauden välein, jolloin todettiin sen hetkiset tilanteet ja sovittiin eri työvaiheiden toteutustavat. Työssä toteutui toimeksiantajan esittämät toiveet ja lopputulos tyydytti toimeksiantajan tarpeet.

Työ oli haastava ja mielenkiintoinen mutta samalla melko työläs. Työ aloitettiin tyhjältä pöydältä ja sisälsi suunnittelusta lähtien kaiken valmiiseen työhön vaadittavan. Koodia syntyi asiakassivustoon hieman alle 600 riviä ja hallintasivustoon noin 2800 riviä.

Opinnäytetyötä työstettäessä ja käytettyihin tekniikoihin perehdyttäessä syntyi paljon mahdollisia ominaisuuksia, joista osa lisättiin vaatimusmäärittelyyn. Välttämättömiin ominaisuuksiin ei lisätty työn aloituksen jälkeen uusia kohtia.

LÄHTEET

Imperva, 2011. SQL Injection: By the Numbers. Www-dokumentti. Saatavissa: http://blog.imperva.com/2011/09/sql-injection-by-the-numbers.html. Luettu 6.11.2012.

McDaniel, A, 2012. HTML5. Hoboken, John Wiley & Sons, Inc.

Meloni, J, 2012. Sams teach yourself PHP, MySQL and Apache All in One. Old Tappan, Pearson Education.

Negrino, T & Smith, D, 2007. JavaScript: tehokas hallinta. Helsinki, Readme.fi.

The PHP Group, 1997–2014. PHP Manual. Www-dokumentti. Saatavissa: http://www.php.net/manual/en/.

VAATIMUSMÄÄRITTELY

Välttämättömät vaatimukset

- Etusivu: alku-, loppu- ja keskitekstit
- Tuotesivut: alku- ja lopputekstit
- Tuoteryhmät: alku- ja lopputekstit
- Tuotteet: nimi, hinta
- Tuotekuvat
- Yhteystiedot ja yhteydenottolomake

Hyödylliset vaatimukset

- Otsikkokuvan vaihtaminen
- Tekstimuotoilut
- Tuotteet: saatavuus- ja alennustiedot
- Kysymykset yhteydenottosivulle

Mahdolliset vaatimukset

- Tuotteiden lajittelu (nimi, hinta, saatavuus jne.)
- Tuotteiden haku
- RSS-uutissyötteet
- Sosiaalisen median liitännäiset
- SSL-suojaus hallintasivustolle
- Perustoiminnot kuvankäsittelyyn tuotekuvien lisäyksessä