

Joni Lehto

VERKKOKAUPAN SUUNNITTELU JA TOTEUTUS

Tietojenkäsittelyn koulutusohjelma

2015



VERKKOKAUPAN SUUNNITTELU JA TOTEUTUS

Lehto, Joni
Satakunnan ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma
Maaliskuu 2015
Ohjaaja: Nieminen, Hans
Sivumäärä: 30
Liitteitä: 0

Asiasanat: verkkokauppa, web-ohjelmointi, tietokanta

Tässä opinnäytetyössä oli aiheena suunnitella ja toteuttaa verkkokauppaohjelmisto. Ohjelmistosta oli tarkoitus tehdä mahdollisimman toimiva, tietoturvallinen ja käyttäjäystävällinen.

Raportin teoriaosuudessa käydään läpi toteutuksessa käytävät tekniikat ja esitetään koodiesimerkkejä niiden käytöstä. Sen jälkeen esitellään projektissa toteutetun sivuston ja tietokannan rakenne. Lopuksi tutkitaan tarkemmin verkkokaupasta löytyvää toiminnollisuutta ja otetaan käsittelyyn niiden ohjelmointikoodia.

Työssä seurattiin sen alussa tehtyä suunnitelmaa ja aikataulutusta. Näin toimimalla saatiin lopputulokseksi haluttu verkkokauppaohjelmisto, joka täyttää sille alussa asetetut vaatimukset.

DESIGNING AND IMPLEMENTING AN ONLINE SHOP

Lehto, Joni

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in Business Information Systems

March 2015

Supervisor: Nieminen, Hans

Number of pages: 30

Appendices: 0

Keywords: online shop, web development, database

The purpose of this thesis was to design and implement an online shop. When planning the software there was a priority on it being as practical, secure and user-friendly as possible.

The theoretical part of this thesis goes through the techniques that are used in the implementation of the software and provides examples of their use. Followed by that is an introduction to the site and database structure. Finally, the functionality of the software is being examined and will be covered with the programming code.

The project followed the initial plan and schedule. By doing this the final result was the desired web shop software that meets all the requirements set out for it at the beginning.

SISÄLLYS

1	JOHDANTO.....	5
2	VERKKOKAUPAN SUUNNITTELU	5
3	KÄYTETTÄVÄT TEKNIIKAT	6
3.1	HTML	7
3.2	CSS	8
3.3	JavaScript.....	9
3.4	PHP	9
3.5	MySQL	10
4	SIVUSTON TIETOKANTA.....	12
5	SIVUSTON RAKENNE	15
5.1	Ulkoasu	15
5.2	Kategoria- ja tuotesivu.....	15
6	SIVUSTON TOIMINNOT.....	17
6.1	Rekisteröityminen	17
6.2	Kirjautuminen	20
6.3	Tuotehaku	21
6.4	Ostoskori.....	21
6.5	Tilauksen tekeminen	22
6.6	Tilinhallinta.....	22
6.6.1	Tietojen muuttaminen.....	23
6.6.2	Käyttäjän tilaukset	24
6.7	Ylläpitäjän hallintapaneeli	25
6.7.1	Käyttäjien hallinta	25
6.7.2	Tilausten hallinta	26
6.7.3	Kategorioiden hallinta	26
6.7.4	Tuotteiden hallinta.....	27
7	YHTEENVETO	28
7.1.1	Suunnittelun arviointi	28
7.1.2	Työn itsearviointi.....	28
	LÄHTEET.....	30

1 JOHDANTO

Verkkokaupat ovat nykyisin vakiintunut osa arkipäiväistä kaupankäyntiä. Kunnollinen verkkokauppaohjelmisto on monelle yritykselle elintärkeä osa yrityksen toimintaa. Tässä opinnäytetyössä oli aiheena suunnitella ja toteuttaa hyvin toimiva ja käyttäjäystävällinen verkkokauppa.

Verkkokauppaohjelmistoon on mahdollista luoda todella paljon eri asiakaskokemusta rikastuttavia ominaisuuksia. Tässä työssä keskityttiin kuitenkin tekemään kokonaisuutena toimiva ja nykyaikainen verkkokauppa, josta löytyy kaikki kaupankäynnin osalta oleelliset toiminallisuudet perusteellisesti toteutettuna.

Opinnäytetyössä esitellään aluksi suunnitelma verkkokaupan toteuttamiseen ja työssä käytettävät tekniikat. Niiden jälkeen tutustutaan sekä itse sivuston, että sen tietokannan rakenteeseen ja lopuksi käydään läpi varsinaista ohjelmointia vaativia ominaisuuksia. Monissa kappaleissa otetaan myös katsaukseen mahdolliset turvallisuuteen liittyvät riskit ja käydään läpi tapoja niiden hoitamiseen.

2 VERKKOKAUPAN SUUNNITTELU

Ennen projektin aloittamista on se suunniteltava huolellisesti. Tähän kuuluu tyypillisesti vaatimusmäärittely. Vaatimusmäärittelyssä kuvataan vaatimukset ohjelmistolle mitä sen valmistuessa siltä voidaan odottaa.

Projektin ainoa työntekijä oli allekirjoittanut, joten hoidin määrittelyn itselleni parhaimmaksi katsomallani tavalla. Koska projektia ei tehty yritykselle, ei sillä myöskään ollut tarkkoja vaatimuksia. Tehtävänäni oli siis ensimmäiseksi selvittää mitä kaikkia ominaisuuksia verkkokauppaan tarvittaisiin. Tutkimalla muutamia verkkokauppoja ulkoisesti sai helposti selvitettyä mitä kaikkia vaatimuksia asiakkailta voisi olla verkkokaupalle. Näiden lisäksi oli kuitenkin mietittävä kaupan ylläpitäjän tarvitsemia ominaisuuksia. Ylläpitäjän hallintaominaisuuksia tuli tutkittua joistakin valmiista verkkokauppaohjelmistoista. Näin sain listattua kymmeniä ominaisuuksia, joi-

den kaikkien toteuttamisessa menisi taatusti aivan liikaa aikaa. Seuraavaksi oli siis karsittava mitkä ominaisuudet voidaan jättää tässä vaiheessa toteuttamatta ohjelmistosta.

Aikatauluksi projektille olin varannut itseltäni yhden kalenterikuukauden, joskin olin valmis venyttämään tätä aikaa tarvittaessa hieman pidempään. Aikataulun perusteella suunnittelin sitten mitä kaikkea minulla on aikaa toteuttaa. Jaoin toteutuksen kolmeen osaan, HTML-sivuston, tietokannan ja sivuston toiminnollisuuden toteuttamiseen.

Ensimmäiseksi oli verkkokaupalle tuli toteuttaa jonkinlainen ulkoasu ja tietokanta. Näiden tekemiseen varasin aikaa yhden viikon. Loput kolme viikkoa olivat sen jälkeen käytettävissä toiminnollisuuden ohjelmointiin. Toiminnot palastelin aikatauluun vielä tarkemmin 2-4 päivän palasiin parhaimman arviointikykyäni perusteella. Näin sain käsitystä mitä kaikkea ehtisin ohjelmistoon toteuttaa aikataulun puitteissa ja jätin vähemmän tarpeellisia toimintoja pois sen perusteella.

Suunnitelmaksi jäi loppujen lopuksi toteuttaa verkkokauppaohjelmisto, josta löytyisi kaikki tarpeelliset ominaisuudet sujuvaan kaupankäyntiin. Painopisteenä oli myös tehdä kaikista toiminnoista mahdollisimman käyttäjystävällisiä ja varmasti turvallisia. Sivuston ulkoasuun ei ollut tarkoitus panostaa tässä projektissa merkittävästi. Ajatukseksi jäi, että ulkoasu olisi kaupan käyttöönottovaiheessa helposti muokattavissa esimerkiksi kaupan teemaan sopivaksi. Myös lisätoimintoja olisi helposti tehtävissä lisää kaupan vaatimusten mukaan.

3 KÄYTETTÄVÄT TEKNIIKAT

Tässä työssä tehty sivusto rakennettiin käyttämällä HTML-merkintäkieltä (Hypertext Markup Language) ja sen ulkoasu toteutettiin CSS-tyylikielillä (Cascading Style Sheets). Lisäksi sivuston käyttökokemusta rikastuttamaan käytettiin JavaScript-ohjelmointikieltä. Varsinaisen toiminnollisuuden lisääminen sivustolle tehtiin PHP-

ohjelmointikieltä (PHP: Hypertext Preprocessor) käyttämällä. Sivuston relaatiotietokantaohjelmistona käytettiin MySQL:ää.

3.1 HTML

HTML on internet-sivustojen luontiin käytettävä merkintäkieli. Sen viimeisin versio on HTML5, jota myös tässä työssä käytettiin. Tämä versio julkaistiin virallisesti W3C:n suositukseksi 28. lokakuuta 2014. Edellinen versio, HTML 4, julkaistiin vuonna 1997. Yhtenä merkittävämpänä uutena ominaisuutena HTML5:ssä on tuki lisätä sivustoille helposti multimediasisältöä <video> ja <audio> -elementtien avulla. Näin ollen tällaista sisältöä ei tarvitse enää esittää erillisten ohjelmien, kuten Adobe Flash Playerin kautta. (W3C 2014.)

Kuvassa 1 on esitetty yksinkertaisen sivun HTML-koodi. Siitä löytyy HTML-sivulle tyypilliset tunnisteet: <!doctype>, <html>, <head>, <title> ja <body>. Jokainen näistä tunnisteista avataan ja suljetaan, sekä niiden väliin lisätään sisältö. Tunnisteen sulkeminen tapahtuu lisäämällä kauttaviiva toisen vastaavan tunnisteen alkuun. Tältä sivulta löytyy myös <p> ja </p> -tunnisteet, joiden välistä löytyy teksti ”Esimerkkitekstiä”. <p>-tunniste tarkoittaa tekstikappaletta. Käytännössä tämä sivu näkyy käyttäjälle selaimessa pelkkänä tekstinä ”Esimerkkitekstiä”.

```
<!doctype html>
<html>
<head>
<title>Otsikko</title>
</head>
<body>

    <p>Esimerkkitekstiä</p>

</body>
</html>
```

Kuva 1. Tekstieditori näyttää HTML-sivun todellisen sisällön

3.2 CSS

CSS on tyylikieli, jota käytetään merkintäkielellä tehtyjen dokumenttien ulkonäön muuttamiseen. Yleinen käyttötarkoitus tälle on HTML-sivustojen tyyllittely. Käytännössä CSS:n avulla voidaan muokata vaikkapa fonttien väriä ja kokoa, tai lisätä taustakuva sivustolle.

CSS-tyyliohjeet voidaan lisätä joko HTML-tiedostoon mukaan, tai sitten luoda niille kokonaan erillinen tiedosto ja lisätä HTML-sivulle ainoastaan viittaus (kuvassa 2) siihen. Erillinen CSS-tiedosto mahdollistaa samojen tyyliohjeiden käyttämisen useilla sivuilla ilman että niitä tarvitsee lisätä jokaiselle HTML-sivulle erikseen. Lähes poikkeuksetta on suositeltavaa käyttää erillistä CSS-tiedostoa.

```
<html>
<head>
<link rel="stylesheet" href="styles.css">
```

Kuva 2. HTML-koodissa voidaan viitata erilliseen CSS-tyylitiedostoon

Seuraavaksi käydään läpi kuvassa 3 oleva yksinkertainen esimerkki CSS-tyyllittelystä. Ensiksi CSS-tiedostossa kerrotaan minkä tunnisteiden sisältöä halutaan muokata ja sen jälkeen lisätään aaltosulkeet. Aaltosulkeiden sisään lisätään itse asetuksia muodossa: muokattava ominaisuus, kaksoispiste, haluttu arvo, puolipiste. Kuvan 3 CSS-tiedostossa asetetaan HTML-sivulta löytyvälle body-tunnisteelle tyyli-määrittelyjä. Tämän elementin sisältö noudattaa nyt seuraavia neljää asetusta: Background-color asettaa sivun taustaväriksi värikoodin ”#ebf0f9”, joka vastaa vaaleansinistä. Color asettaa fontin väriksi värikoodin ”#000000”, joka vastaa mustaa. Font-size asettaa fontin kooksi 12 pikseliä. Width asettaa elementin leveydeksi 960 pikseliä.

```
body {
  background-color: #ebf0f9;
  color: #000000;
  font-size: 12px;
  width: 960px;
}
```

Kuva 3. CSS-tyylissä määritellään HTML-sivulle ulkonäköasetuksia

3.3 JavaScript

JavaScript on ohjelmointikieli, jonka avulla internet-sivustoille voi lisätä dynaamista sisältöä. JavaScript-koodi suoritetaan sivuston käyttäjän omalla tietokoneella, eikä vaikkapa sivuston palvelinkoneella.

JavaScriptin käyttöä nopeuttamaan ja helpottamaan voi käyttää erilaisia kirjastoja, kuten jQuery, jota tässäkin työssä käytettiin. jQuery tarjoaa helpon tavan esimerkiksi käsitellä elementtejä ja tapahtumia, sekä luoda animaatioita. Nämä kaikki toimivat lisäksi suoraan kaikilla suosituimmilla selaimilla, poistaen tarpeen sivuston kehittäjältä luoda ne manuaalisesti jokaiselle.

3.4 PHP

PHP on ohjelmointikieli, joka on suunnattu erityisesti webohjelmointiin. JavaScriptin tavoin PHP:n avulla sivustolle voidaan lisätä toimintoja. Yhtenä isona erona JavaScriptiin on kuitenkin se, että PHP-koodi suoritetaan palvelimella. Näin ollen PHP-koodi ei ole myöskään näkyvillä sivuston vierailijoille tutkimalla HTML-tiedostoa tekstieditorissa.

W3Techsin (2015) tekemän tutkimuksen mukaan PHP on suosituin websivuilla käytettävä palvelinpään ohjelmointikieli 82,0 % markkinaosuudella 1. päivä tammikuuta 2015. Seuraavaksi suosituimmalla kilpailijalla, Microsoftin ASP.NET:lla, ollessa 17,1 % markkinaosuus. PHP-kielen suureen käyttäjämäärään vaikuttaa varmasti sen ilmaisuus. Toisin kuin ASP.NET, PHP tarjoaa ilmaisen ohjelmiston käytettäväksi sekä harrastus-, että yrityskäyttöön.

PHP-koodia voidaan lisätä HTML-sivun sekaan laittamalla se ”<?php” -aloitustunnisteen ja ”?>” -lopetustunnisteen väliin, kuten kuvassa 4. Tällöin palvelinohjelmistossa oleva tulkki suorittaa näiden tunnisteiden välissä olevan koodin sivun ladatessa. Kuvan 4 PHP-koodissa luodaan ensin muuttuja \$ika, jonka arvoksi

asetetaan 15. Tämän jälkeen if-ohjausrakenteella tutkitaan onko muuttujassa oleva pienempi kuin 18. Mikäli tämä ehto ei pidä paikkaansa, eli muuttujan arvo on vähintään 18, tulostetaan merkkijono ”Henkilö ei ole täysi-ikäinen.” Tämä merkkijono näytetään sitten websivulla p-tunnisteiden välissä.

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Otsikko</title>
</head>
<body>

<p>
<?php
$ika = 15;

if ($ika < 18) {
    echo "Henkilö ei ole täysi-ikäinen"
}

?>
</p>

</body>
</html>
```

Kuva 4. PHP-koodia voidaan lisätä kätevästi HTML-koodin sekaan

3.5 MySQL

MySQL on Oracle Corporationin avoimen lähdekoodin relaatiotietokantaohjelmisto. Sen avulla voi tallentaa, hakea ja käsitellä tietokannan tietoa. MySQL:n käyttö on ilmaista, joskin ohjelmasta on tarjolla myös maksullisia versioita yrityskäyttöön suunnattuna. MySQL on saavuttanut maailmalla kovan suosion ja sen asiakkaita ovatkin monet maailman suurimmista yrityksistä, kuten Facebook, Google, Wikipedia ja YouTube. (MySQL.)

Monista ohjelmointikielistä, kuten myös PHP:stä löytyy MySQL:lle valmis kirjasto, jonka avulla sitä voi käyttää helposti. Kuvassa 5 näkyy miten PHP-koodissa voi yk-

sinkertaisesti yhdistää ja kysellä MySQL-tietokantaa käyttäen mysqli-kirjastoa. Aluksi muodostetaan yhteys tietokantaan käyttäen mysqli_connect-lausetta, johon annetaan parametreiksi tietokantapalvelimen IP-osoite, käyttäjätunnus tietokannassa, käyttäjätunnuksen salasana ja käytettävä tietokanta. Yhdistämisen jälkeen tietokannasta kysellään tietoja käyttäen mysqli_query-lausetta, johon annetaan parametreiksi tietokantayhteyden sisältävä muuttuja ja itse SQL-lause. Tässä tapauksessa käytetään SELECT-lausetta nimen ja iän kyselyyn. Saadut tiedot esitetään sitten HTML-sivulla.

```
$link = mysqli_connect("127.0.0.1", "user", "pw", "db");

if (!$data) {
    die(mysqli_error($link));
}

?>

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Otsikko</title>
</head>
<body>

<?php

$sql = "SELECT name, age FROM customers WHERE id = 23";

$data=mysqli_query($link, $sql);

if (!$data) {
    die(mysqli_error($link));
}
else {
    echo ($data['name'] . "<br>" . $data['age']);
}

?>

</body>
</html>
```

Kuva 5. PHP:sta löytyy valmis MySQL-kirjasto nopeuttamaan sen käyttöä webohjelmistoissa

4 SIVUSTON TIETOKANTA

Tässä työssä tietokannan hallintaan käytetään MySQL-ohjelmistoa. Sivuston tietokanta sisältää yritykselle elintärkeää tietoa esimerkiksi asiakkaista, tilauksista ja tuotteista. Kuvassa 6 on esiteltyä sivuston tietokannan rakenne. Tietokanta sisältää viisi taulua, jotka käydään seuraavaksi tarkemmin läpi.

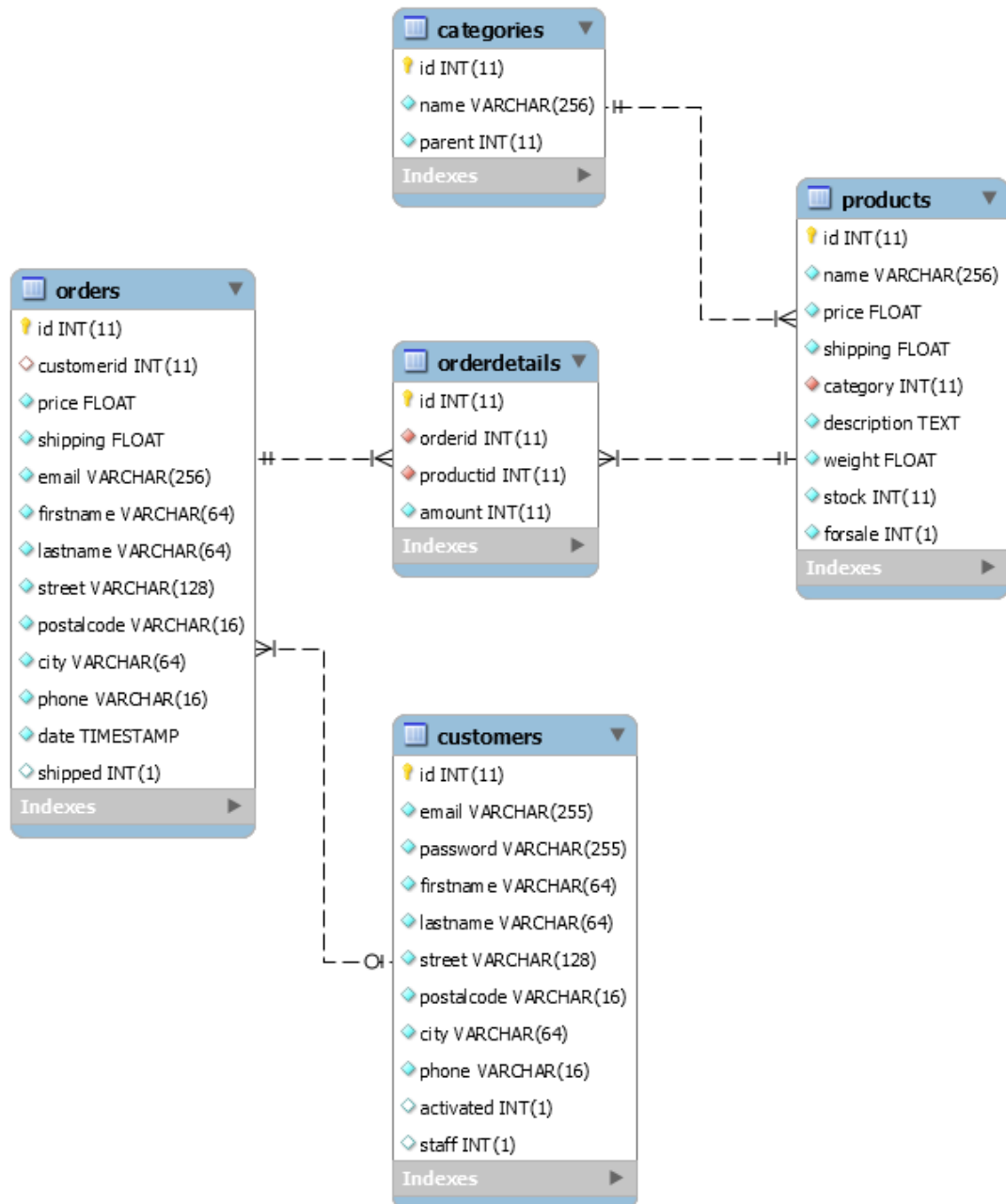
Customers-taulusta löytyy verkkokaupan asiakkaiden tiedot. Se sisältää esimerkiksi id-numeron, sähköpostiosoitteen, salasanan tiivisteen, nimen, osoitteen ja muita tarpeellisia tietoja. Id-numero on jokaiselle asiakkaalle luotu uniikki numerosarja, jota hän ei tyypillisesti itse näe missään. Sitä käytetään muun muassa sivuston toiminnollisuuteen liittyvissä asioissa. Sähköpostiosoite toimii sisäänkirjautumistunnuksena sivustolle. Se on uniikki tunnus, jonka asiakas yleensä myös muistaa. Toinen yleinen tapa toteuttaa sisäänkirjautuminen sivustolle olisi antaa asiakkaan itse keksiä oma tunnuksensa. Salasana on tallennettu tietokantaan salasanatiivisteinä. Tämä tarkoittaa, että salasana ei ole luettavissa selkokielellä, vaan ainoastaan siitä tehty tiiviste on luettavissa. Käyttäjän kirjautuessa sivustolle, hänen syöttämästä salasanasta tehdään aina tiiviste ja tätä tiivistettä verrataan tietokannassa olevaan tiivisteeseen. Näin ollen edes sivuston ylläpitäjä ei näe käyttäjien salasanvoja selkokielellä. Tästä syystä myöskään salasanan palauttaminen käyttäjälle sen unohtuessa ei ole mahdollista, vaan on aina luotava uusi salasana vanhan unohtuessa.

Orders-taulu sisältää verkkokaupan tilaukset. Siitä löytyy muun muassa tilauksen id-numero, asiakkaan id-numero, hinta, postikulut, tilauksen status sekä asiakkaan nimi ja osoite -kentät. Vaikka taulusta löytyykin asiakkaan id-numero, jonka avulla hänen osoitteensa voitaisiin noutaa hänen profiilistaan, täytyy tilauksen oma osoite aina tallentaa, koska asiakas voi haluta toimituksen eri osoitteeseen kuin hänen kotiosoitteensa. Asiakkaan id-numeroa käytetäänkin yhdistämään tilaukset hänen profiiliinsa. Näin ollen hän voi selvittää omat tilauksensa ja niiden statukset omasta profiilistaan helposti. Tilauksen status-kenttä on oletuksena luku 0. Tämä tarkoittaa tilauksen olevan vastaanotettu verkkokaupan puolelta, mutta sille ei ole vielä tehty mitään. Statuksia voidaan lisätä haluttu määrä, jolloin sekä asiakkaan, että kauppiaan on helppo seurata sen etenemistä tarkasti. Tällaisia statuksia voisi olla vaikkapa ”maksettu”, ”toimituksessa” ja ”toimitettu”.

Categories-taulu sisältää verkkokaupan kategoriat. Tästä taulusta löytyy id-numero, nimi ja isäntäkategoria -kentät. Id-numeron perusteella kategoriat voidaan tunnistaa. Nimi-kentästä löytyy kategorian nimi, joka näytetään sivuston kategoriavalikossa. Isäntäkategoria-kentässä oleva numero voi olla joko luku 0, jolloin kategorialla ei ole isäntää, tai sitten yli 0:n oleva kokonaisluku, jolloin tämän kategorian isäntäkategoria on tällä id-numerolla löytyvä kategoria. Myös isäntäkategorialla voi olla isäntäkategorioita.

Products-aulussa on verkkokaupan tuotteet. Taulusta löytyy muun muassa id-numero, nimi, hinta, kuvaus, varastosaldo, myynnissä ja kategoria -kentät. Id-numeron perusteella tuotteet voidaan tunnistaa. Nimi-kentässä on tuotteet nimi ja kuvaus-kentässä on tuotteen kuvaus. Nämä tiedot esitetään verkkokaupassa tuotteen tiedoissa. Hinta-kentästä löytyy tuotteen hinta. Varastosaldo-kentässä on myynnissä olevien kappaleiden määrä. Myynnissä-kentän avulla verkkokaupan ylläpitäjä voi piilottaa kukin tuotteen sivun asiakkailta. Kategoria-kentässä on kategorian id-numero, johon kyseinen tuote halutaan sijoittaa verkkokaupassa.

Orderdetails-taulun avulla tilauksiin yhdistetään tiedot tilatuista tuotteista ja niiden määristä. Taulusta löytyy id-numero, tilauksen id-numero, tuotteen id-numero ja määrä -kentät. Yhdessä tilauksessa voi olla useita tuotteita ja niiden määrä voi olla useampi kuin yksi. Tätä taulua käytetään näiden tietojen yhdistämiseen. Tilauksen id-numero kertoo mitä tilausta kyseisen rivin tiedot koskevat. Tuotteen id-numero kertoo mitä tuotetta tilauksesta löytyy. Määrä-kentässä on tilauksessa olevan tuotteen kappaleiden määrä. Yhdellä tilauksella löytyy siis yksi rivi tästä taulusta jokaista tuotetta kohtaan, joka tilauksessa on.

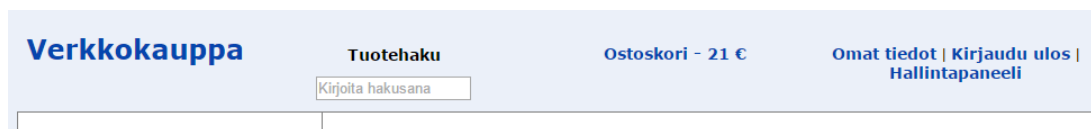


Kuva 6. Verkkokaupan tietokannan rakenne

5 SIVUSTON RAKENNE

5.1 Ulkoasu

Sivuston ulkoasusta oli tarkoitus tehdä mahdollisimman selkeä ja käyttäjäystävällinen. Yksi tapa toteuttaa näitä ehtoja oli tuoda tärkeimmät linkit selkeästi esille. Kuvassa 7 on näkyvillä sivuston yläosan linkkien sijoittelu. Vasemmassa yläkulmassa on esillä verkkokaupan nimi, jota painamalla pääsee etusivulle. Sen vierestä löytyy tuotehaku, ostoskori, kirjautuminen, rekisteröinti, käyttäjätilinhallinta ja ylläpitäjän hallintapaneeli. Sivun vasemmassa reunassa on lohko, josta löytyy verkkokaupan tuotekategoriat ja sen oikealla puolella sisältölohko, johon tuotetaan kunkin sivun sisältö. Eli esimerkiksi ostoskori-linkkiä klikkaamalla sisältölohkoon tulee esille tiedot ostoskorin sisällöstä. Sivun alareunasta löytyy vielä kaupan yhteystiedot ja linkkejä vähemmän tärkeisiin tietoihin.



Kuva 7. Verkkokaupan tärkeimmät linkit ovat aina näkyvillä jokaisen sivun yläosassa

Varsinaista grafiikkaa ei ulkoasua varten tehty, vaan se on toteutettu kokonaan käyttämällä HTML-, CSS ja JavaScript-koodia. Tämä tarkoittaa, että sivuston ulkoasu on myös helposti muokattavissa, joka oli myös tarkoituksena. Mikäli esimerkiksi verkkokaupan taustaväriä halutaan vaihtaa, tämä voidaan tehdä yksinkertaisesti muokkaamalla styles.css -tiedostossa olevaa "background-color"-kentän arvoa. Verkkokauppaohjelmistoa voi näin ollen soveltaa moneen eri teemaan muokkaamalla pelkkää koodia, eikä uutta graafista ulkoasua ole aina pakko luoda.

5.2 Katteoria- ja tuotesivu

Jokaisella tuotteella on oma sivunsa verkkokaupassa. Näitä sivuja ei kuitenkaan ole onneksi tarvetta lisätä käsin jokaiselle tuotteelle, vaan ne voidaan tuottaa dynaami-

sesti PHP-koodilla. Käytännössä on olemassa vain yksi tuotesivu, jolle haetaan aina halutun tuotteen tiedot tietokannasta. Tuotesivu sisältää staattisena koodina taulukon, jossa on sarakkeina tuotteen tietoja, kuten nimi, hinta ja varastosaldo. Näiden alle sitten sijoitetaan PHP-koodissa kyselyyn tuotteen omat tiedot, kuten kuvassa 8.

```
<h2>Tuotteen tiedot</h2>
```

```
<table>
  <tr>
    <th>Nimi</th>
    <th>Hinta</th>
    <th>Toimituskulut</th>
    <th>Paino</th>
    <th>Varasto</th>
  </tr>
```

```
<?php
$sql = "SELECT * FROM products WHERE id=" . $_GET["id"];
$data=mysqli_query($link, $sql);
if (!$data) die(mysqli_error($link));
$rows=mysqli_num_rows($data);
for ($i = 0; $i < $rows; $i++)
{
    $rivi=mysqli_fetch_assoc($data);
    echo('<tr>');
    echo('<td>' . $rivi['name'] . '</td>');
    echo('<td>' . $rivi['price'] . '</td>');
    echo('<td>' . $rivi['shipping'] . '</td>');
    echo('<td>' . $rivi['stock'] . '</td>');
    echo('</tr>');
}
?>
</table>
```

```
<h2>Kuvaus</h2>
```

```
<?php
echo('<td>' . $rivi['description'] . '</td>');
?>
```

Kuva 8. HTML-koodin seassa voi olla PHP-koodia, joka tuottaa dynaamista sisältöä sivulle

Kategoriat näkyvät sivuston vasemmassa osassa ja jokainen tuote löytyy jonkin näiden alta. Kategorioiden avulla asiakkaiden on tarkoitus löytää tuotteet helpommin kuin vaikkapa selaamalla yksittäistä listausta, jossa olisi kaikki verkkokaupan tuotteet. Kategorioiden käyttö alkaa olemaan melkein pakollista kun tuotteita alkaa olemaan useita kymmeniä.

Tuotteiden järjestäminen kategorioihin on toteutettu verkkokaupassa siten, että products-aulussa on pakollinen category-sarake, johon sijoitetaan kategorian id-numero. Se kertoo minkä kategorian alta tuote löytyy. Koska category-sarake on asetettu tietokannassa pakolliseksi, on sellainen oltava jokaisella tuotteella.

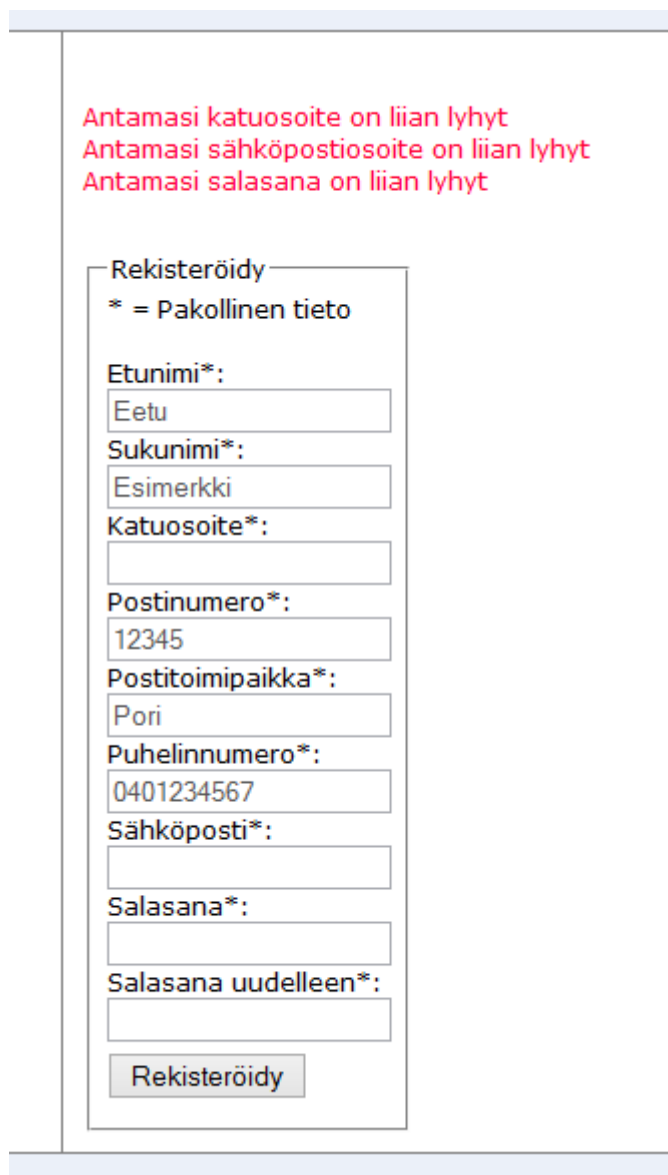
Kuten myös luvussa 4 kerrotaan, jokaisella kategorialla voi olla isäntäkategoria. Isäntäkategorioita voi olla lisäksi useita kerroksia. Esimerkki tällaisista kerroksista voisivat olla vaikkapa seuraava: tietokoneet, komponentit, näytönohjaimet. Eli Tietokoneet-kategorialla on Komponentit-lapsikategoria, jolla on vielä oma Näytönohjaimet-lapsikategoriensa.

6 SIVUSTON TOIMINNOT

6.1 Rekisteröityminen

Rekisteröityminen on yksi yleisimmistä ominaisuuksista verkkokaupassa. Työssäni rekisteröinnistä (kuvassa 9) oli tarkoitus tehdä mahdollisimman helppokäyttöinen, jotta asiakkaan kynnys luoda tunnus verkkokauppaan olisi hyvin alhainen. Sivuston ylälaudassa on linkki rekisteröinti-sivulle. Painamalla kyseistä linkkiä asiakas pääsee syöttämään tarvittavat tiedot käyttäjätunnuksen luotiin. Rekisteröintiin tarvittavat tiedot ovat sähköposti-osoite, nimi, osoite, puhelinnumero ja salasana. Kuten myös luvussa 4 kerrotaan, rekisteröintiin ei käytetä erillistä käyttäjätunnusta, vaan sähköpostiosoite toimii tunnuksena kirjautuessa. Salasana varmistetaan pyytämällä se kah-

teen kertaan ja vertaamalla annettuja merkkijonoja. Mikäli ne eivät täsmää, annetaan virheilmoitus.



Antamasi katuosoite on liian lyhyt
Antamasi sähköpostiosoite on liian lyhyt
Antamasi salasana on liian lyhyt

Rekisteröidy
* = Pakollinen tieto

Etunimi*:

Sukunimi*:

Katuosoite*:

Postinumero*:

Postitoimipaikka*:

Puhelinnumero*:

Sähköposti*:

Salasana*:

Salasana uudelleen*:

Kuva 9. Rekisteröintisivu antaa palautetta tekstikentistä, joissa on puutteelliset tiedot

Käyttäjän antaessa salasanan, tästä luodaan tiiviste käyttämällä PHP:sta löytyvää vahvaa `password_hash`-funktiota. Tämä funktio lisättiin PHP:hen versiossa 5.5, joka julkaistiin kesäkuussa 2013 (PHP, 2013). Ennen tätä funktiota salasana tiivisteen tekeminen ja suolaus täytyi toteuttaa erillisillä funktioilla. Salasan suolaaminen tarkoittaa, että siihen liitetään joitakin muita merkkejä ennen kuin siitä luodaan tiiviste.

Tämän tarkoituksena on tehdä salasanaatiivisten murtamisesta merkittävän paljon haastavampaa.

Käyttäjän täytettyä tiedot lomakkeeseen ja painaessa rekisteröidy-painiketta, lähetetään tiedot palvelimelle käyttämällä POST-metodia. POST-metodi on asiakkaan selaimen lähettämä pyyntö, jossa on tyypillisesti tarkoituksena lähettää jotakin tietoa palvelimelle. Toinen yleinen metodi on GET, jossa taas ei tyypillisesti lähetetä mitään tietoa, joskin se on mahdollista.

Kaikessa käyttäjältä saatavissa tiedoissa on oltava tarkkana tietoturvan suhteen. Rekisteröintisivulla käyttäjältä saadut tiedot ajetaan ensin PHP:n filter_input-funktion lävitse. Tämän funktion avulla käyttäjän syötteistä voidaan poistaa mahdollisia epähaluttuja merkkejä ja merkkijonoja. Lisäksi syötteitä lisättäessä tietokantaan käytetään kuvassa 10 esiteltävää Prepared Statement -tekniikkaa. Tässä tekniikassa itse SQL-lause ja tiedot lähetetään SQL-palvelimelle erikseen. Tämän tekniikan tarkoitus on estää mahdolliset SQL-injektiot.

```
// Valmistellaan uusi asiakas lisättäväksi customers-tauluun
$sql=$link->prepare("INSERT INTO customers
(email, password, firstname, lastname, street, postalcode, city, phone)
VALUES (?, ?, ?, ?, ?, ?, ?, ?)");

// Asetetaan SQL-lauseeseen muuttujiin suodatetut tiedot
$sql->bind_param('sssssss', $email, $passwordhash, $firstname, $lastname,
                $street, $postalcode, $city, $phone);

// Suoritetaan SQL-lause
if ($sql->execute())
{
```

Kuva 10. Tietokantaan lisättävät tiedot valmistellaan lisättäväksi Prepared Statement -tekniikalla

SQL-injektio on hyökkäystapa sivuston tietokantaan, jossa sinne yritetään lähettää vahingollisia SQL-lauseita. Näin hyökkääjä saattaa päästä käsiksi esimerkiksi tietokannassa oleviin tietoihin ilman oikeita käyttöoikeuksia. Yleinen tapa aiheuttaa SQL-injektio on kokeilla lähettää SQL-komentoja verkkosivustolla oleviin lomakkeisiin ja tutkia onko SQL-palvelinta mahdollista saada suorittamaan näitä komentoja.

6.2 Kirjautuminen

Rekisteröinnin lisäksi tarvitaan tietysti kirjautuminen. Kirjautumissivulla asiakas syöttää lomakkeeseen sähköpostinsa ja salasansa. Kuten rekisteröinnissäkin, myös tällä sivulla käyttäjän syöttämästä salasanasta luodaan tiiviste. Tällä kertaa luotua tiivistettä verrataan jo tietokannassa olevaan tiivisteeseen. Kyseistä toimenpidettä varten PHP:stä löytyy `password_verify`-funktio, johon annetaan parametreiksi yksinkertaisesti käyttäjän syöttämä salasana ja tietokannasta haettu tiiviste, kuten kuvassa 11 näkyy. Funktio palauttaa sitten joko `true` tai `false` -arvon sen perusteella täsmäisivätkö salasanatiivisteet. Mikäli annettu salasana on oikein, lisätään käyttäjän istuntoon hänen käyttäjätunnuksensa ja ohjataan hänet etusivulle, tai vaihtoehtoisesti annetaan virheilmoitus väärästä salasanasta. Kirjautuneena ollessa käyttäjän sisäänkirjautumislinkki korvataan uloskirjautumiseen johtavalla linkillä. Uloskirjautumisessa yksinkertaisesti käyttäjän istunto poistetaan ja hänet ohjataan verkkokaupan etusivulle.

```

if (isset($_POST['email'], $_POST['password'])) {
    $email=filter_input(INPUT_POST, 'email', FILTER_SANITIZE_EMAIL);
    $password=filter_input(INPUT_POST, 'password', FILTER_SANITIZE_STRING);

    $sql=$link->prepare("SELECT id, password, staff FROM customers WHERE email = ?");
    $sql->bind_param("s", $email);
    $sql->execute();
    $sql->bind_result($id, $dbpassword, $staff);
    $sql->fetch();

    if (password_verify($password, $dbpassword)) {
        $_SESSION["id"] = $id;
        $_SESSION["staff"] = $staff;
        header("Location: /index.php");
    }
    else {
        echo("<p class='error'>Kirjautuminen epäonnistui. Yritä uudelleen.</p>");
    }
}

```

Kuva 11 Kirjautumisessa salasanasta luodaan tiiviste ja sitä verrataan tietokannassa olevaan tiivisteeseen

Session eli istunto on PHP:n tekniikka säilyttää tietoa väliaikaisesti. Näitä tietoja ei siis tallenneta pysyvästi tietokantaan, vaan ne poistetaan yleensä sen jälkeen kun käyttäjä poistuu sivustolta tai kirjautuu ulos. Istunnossa säilytettävät tiedot voivat olla esimerkiksi asiakkaan käyttäjätunnus tai hänen ostoskorissansa olevat tuotteet.

Kirjautumissivulta löytyy myös lomake unohtuneen salasanan uusimiseen. Asiakas voi tilata linkin uuden salasanan asettamiseen sähköpostiinsa. Linkistä avatulla sivulla asiakas syöttää haluamansa salasanan kahdesti aivan kuten rekisteröinnissäkin ja mikäli syötetyt salasanat täsmäävät, korvataan tietokannassa oleva nykyinen salasanatiiviste uudesta salasanasta luodulla tiivisteellä.

6.3 Tuotehaku

Tuotteiden haku on oleellinen osa nopeuttamaan verkkokaupan käyttöä asiakkaalle. Hakukenttä on näkyvillä jokaisella sivulla sivuston ylälaudassa. Haku toimii niin, että käyttäjä syöttää hakuehdon kenttään. Tämän jälkeen PHP-koodissa lähetetään kysely tietokannan products-tauluun, jossa WHERE-ehdossa annetaan käyttäjän antama merkkijono. Kysely palauttaa kaikki rivit joissa tuotteen nimessä esiintyy annettu merkkijono. Palautetuista riveistä esitetään tuotteiden nimet sitten asiakkaalle sivustolla ja annetaan linkit tuotesivuille käyttämällä rivistä löytyvää id-saraketta linkissä. Kuten muutkin asiakkaan syötteet, myös haku on suojattu SQL-injektioilta.

6.4 Ostoskori

Ostoskorin kautta asiakkaat voivat tilata haluamansa tuotteet verkkokaupasta. Jokaisella tuotesivulla on ”Lisää ostoskoriin” -linkki, jota painamalla tuotteen id-numero lisätään asiakkaan istuntoon ja sen määräksi asetetaan 1. Ostoskorin sisältö säilyy siis ainoastaan sen ajan kun asiakas on sivustolla, eikä sitä tallenneta palvelimen tietokantaan. Sivuston ylälaudassa on linkki ostoskoriin ja sen sisältämien tuotteiden yhteenlaskettu hinta. Tästä hinnasta asiakas voi helposti päätellä mitä tuotteita hän on jo lisännyt ostoskoriin joutumatta menemään itse ostoskorisivulle.

Ostoskorisivulla asiakas näkee listauksen sinne lisäämistään tuotteistaan ja niiden hinnoista. Kuvassa 12 on näkyvillä ostoskori johon on lisätty kaksi eri tuotetta. Tällä sivulla hän voi myös päivittää tilauksessa olevien tuotteiden määriä. Tuotteet voi poistaa ostoskorista asettamalla niiden määräksi 0.

Tuotehaku
Ostoskori - 800 €

Ostoskori

Nimi	Määrä	Hinta	Yhteensä
Näytönohjain	2	300 €	600 €
Proessori	1	200 €	200 €
Loppusumma: 800 €			

Kuva 12. Ostoskorissa asiakas näkee listauksen sinne lisäämistään tuotteista

6.5 Tilauksen tekeminen

Ostoskorisivulla olevasta Kassalle-painikkeesta asiakas voi siirtyä tilaamaan ostoskorissa olevat tuotteet. Tilaussivulla näytetään oletuksena rekisteröityneen asiakkaan profiiliin tallennetut osoitetiedot. Mikäli asiakas ei ole kirjautunut, nämä kohdat ovat tyhjiä. Tilausten tekeminen onnistuu myös ilman rekisteröintiä verkkokauppaan. Kun yhteystiedot ovat täytetty, voi tilauksen lähettää. Lähetetyt tilaukset lisätään tietokannan orders-tauluun ja niiden sisältämien tuotteiden tiedot orderdetails-tauluun.

6.6 Tilinhallinta

Tilinhallintasivulta asiakas voi tutkia ja hallita omaan profiiliinsa liittyviä tietoja. Tällä sivulla näytetään siis tietokannasta löytyviä tietoja, joissa käyttäjän id-numero täsmää tietokannan tauluista löytyvän customerid-numeron kanssa.

6.6.1 Tietojen muuttaminen

Omat tiedot -sivulla (kuvassa 13) asiakas voi vaihtaa hänen rekisteröinnissä antamiin yhteystietoja. Sivulla on lomake, johon haetaan sivun latauksessa asiakkaan tiedot tietokannan customers-työkalusta. Nämä tiedot asetetaan lomakkeessa oleviin tekstikenttiin oletusarvoiksi. Asiakkaan muutettua arvoja ja painaessa tallenna-painiketta uudet tiedot lähetetään POST-metodilla palvelimelle. Kuten muutkin käyttäjän syötteet, myös nämä tiedot päivitetään tietokantaan käyttäen Prepared Statement -tekniikkaa.

Tällä sivulla voit vaihtaa yhteystietojasi.

Yhteystiedot	
Sähköpostiosoite	<input type="text" value="eetu.esimerkki@esime"/>
Etunimi	<input type="text" value="Eetu"/>
Sukunimi	<input type="text" value="Esimerkki"/>
Katuosoite	<input type="text" value="Jokutie 123"/>
Postinumero	<input type="text" value="12345"/>
Kaupunki	<input type="text" value="Pori"/>
Puhelinnumero	<input type="text" value="0401234567"/>

[Vaihda salasanaa](#)
[Tilaukset](#)

Kuva 13. Tilinhallinnassa asiakas voi vaihtaa tallennettuja yhteystietojaan

Nämä tiedot esitetään oletuksena tilausta tehdessä nopeuttamaan ostamista verkko-kaupasta. Asiakas voi vaihtaa myös salasanaansa painamalla ”vaihda salasana” -linkkiä. Tämä linkki avaa sivun, jolla on lomake, joka täytetään rekisteröinti-sivun tavoin kirjoittamalla sama salasana kahdesti.

6.6.2 Käyttäjän tilaukset

Tehdyt tilaukset -sivulla asiakas voi nähdä tekemänsä tilaukset. Tällä sivulla listataan kaikki orders-taulusta löytyvät rivit, joissa customerid vastaa kirjautuneen käyttäjän id-numeroa. Näistä riveistä luodaan PHP:lla HTML-tilaukko, jossa on yhtä monta riviä kuin SQL-kyselyn tuloksessa. Jokaisessa tilauksessa on myös linkki kyseisen tilauksen tietoihin.

Tilauksen tiedoissa on nähtävillä kaikki tilauksen tuotteet, kappalemäärät ja niiden hinnat. Tilauslistauksen avulla asiakas voi helposti tarkistaa esimerkiksi hänen hetki sitten tekemänsä tilauksen statuksen. Tilauksen tiedot saadaan selville käyttämällä liitosta SQL-kyselyssä, yhdistämään orders-tilaukko ja orderdetails-tilaukko tilauksen id-numeron perusteella, kuten kuvassa 14. Näiden lisäksi on yhdistettävä vielä mukaan products-tilaukko, jotta tuotteista saadaan niiden tiedot mukaan vastaukseen. Tämä tapahtuu yhdistämällä products-tilaukossa oleva tuotteen id-numero orderdetails-tilaukossa olevan vastaavan kanssa. Kyselyyn on myös lisättävä tieto minkä tilauksen tiedot halutaan saada selville. Tämä tilausnumero saadaan selville GET-metodilla lähetettyä tiedosta.

```
$sql = "SELECT *  
      FROM orders JOIN orderdetails ON orders.id=orderdetails.orderid  
      JOIN products ON orderdetails.productid=products.id  
      AND orders.id=" . $_GET['o'];
```

Kuva 14. SQL-kyselyssä tauluja voidaan yhdistää toisiinsa liitoksella

SQL-kielessä liitos (JOIN) on tapa yhdistää kahden tai useamman taulun rivit. Taulujen liitos tapahtuu yhdistämällä jokin molemmista tauluista löytyvä sarake. Erilaisia liitoksia on SQL-kielessä useita. Erona näissä on se, millä perusteella ne palauttavat rivejä. MySQL:n oletusliitos on INNER JOIN, jota myös kuvan 6 SQL-kyselyssä käytetään. INNER JOIN palauttaa kaikki rivit, joissa on ainakin yksi sama arvo molemmissa tauluissa. Muita liitostapoja ovat esimerkiksi LEFT JOIN ja RIGHT JOIN. LEFT JOIN palauttaa kaikki rivit vasemmasta taulusta ja ne rivit joissa on haluttu arvo oikeasta taulusta. Vastaavasti RIGHT JOIN palauttaa kaikki rivit oikeasta taulusta ja ne rivit joissa on haluttu arvo vasemmasta taulusta. Vasen ja oikea kertovat

kummasta taulusta on kyse ajattelemalla JOIN-sanan olevan keskikohta SQL-lauseessa.

6.7 Ylläpitäjän hallintapaneeli

Ylläpitäjän hallintapaneelista verkkokaupan hallitsija pystyy hoitamaan ylläpidollisia tehtäviä, kuten tilausten- ja tuotteidenhallinta. Erillistä kirjautumista hallintapaneeliin ei ole, vaan customers-taulusta löytyvä arvo ”staff” määrittää onko käyttäjätunnuksella oikeuksia päästä hallintapaneeliin. Hallintapaneeli on tavallisille asiakkaille näkymätön osuus. Linkki sinne näytetään sivustolla ainoastaan silloin, jos kirjautunut käyttäjä on merkitty ylläpitäjäksi tietokannassa.

Vaikka näille sivuille onkin oikeudet ainoastaan sivuston ylläpitäjällä, ei silti ole syytä jättää mahdollisten SQL-hyökkäysten estämistä suorittamatta. Eli tietokantaa käytettäessä tiedot suodatetaan ja valmistellaan aivan kuten vaikkapa rekisteröintisivulla.

6.7.1 Käyttäjien hallinta

Käyttäjien hallintasivulla (kuvassa 15) ylläpitäjä näkee listauksen kaikista verkkokauppaan rekisteröityneistä asiakkaista. Jokaisen asiakkaan asiakasnumerosta klikkaamalla pääsee katsomaan sen asiakkaan kaikki tietokannasta löytyvät tiedot, kuten yhteystiedot, tehdyt tilaukset ja kirjautumisajankohdat. Kaikki nämä tiedot ovat myös muokattavissa tältä sivulta, esimerkiksi siinä tilanteessa, että asiakas ei pysty itse muuttamaan niitä. Käyttäjän tiedoista ylläpitäjä voi myös asettaa käyttäjätunnuksen epäaktiiviseksi, jolloin sillä ei pääse enää kirjautumaan verkkokauppaan.

Käyttäjalista				
Asiakasnumero	Sähköposti	Nimi	Aktivoitu	Ylläpitäjä
1	eetu.esimerkki@esimerkki	Eetu Esimerkki	Kyllä	Kyllä
2	antti.esimerkki@esimerkki	Antti Esimerkki	Kyllä	Kyllä
3	teppo.esimerkki@esimerkki	Teppo Esimerkki	Kyllä	Kyllä
4	heikki.esimerkki@esimerkki	Heikki Esimerkki	Kyllä	Kyllä
5	mauno.esimerkki@esimerkki	Mauno Esimerkki	Kyllä	Kyllä

Kuva 15. Käyttäjien hallinnan kautta verkkokaupan ylläpitäjän on helppo hallita asiakkaitansa

6.7.2 Tilausten hallinta

Tilausten hallintasivulla on nähtävillä listaus kaikista verkkokaupasta tehdyistä tilauksista. Yleisnäkymässä näytetään tilausnumero, tilaajan nimi, summa ja tilauksen tila. Tilaukset voi järjestää näytettäväksi niiden tilan perusteella, eli esimerkiksi ylläpitäjä voi valita näytettäväksi ainoastaan ne tilaukset, jotka ovat tilassa ”maksettu, odottaa käsittelyä.” Tilausnumeroita klikkaamalla pääsee näkemään tilauksen tarkemmat tiedot.

Yksittäisen tilauksen sivulla on näkyvillä tilaajan tiedot, tilatut tuotteet, kappalemäärät, hinnat, tilausajankohta ja tilauksen tila. Tilaa voi päivittää valitsemalla statusnumero listalta ja painamalla Päivitä-painiketta. Tilan muutoksista voidaan myös lähettää asiakkaalle sähköposti. Näin asiakas saa saman tien ilmoituksen, kun tilaus on lähetetty postitettavaksi.

6.7.3 Kategorioiden hallinta

Kategorioiden hallintasivulla ylläpitäjä näkee listauksen verkkokaupan kategorioista. Listauksesta löytyy kategorian nimi, id-numero ja mahdollisen isäntäkategorian id-numero. Id-numeroa painamalla pääsee jokaisen kategorian omalle hallintasivulle. Tältä sivulta voi vaihtaa kategorian nimeä ja isäntäkategorian id-numeroa.

Hallintasivulla voi myös lisätä uusia kategorioita verkkokauppaan. Tämä tapahtuu painamalla ”Lisää uusi kategoria” -painiketta, jolloin aukeaa uusi sivu. Tällä sivulla olevaan lomakkeeseen ilmoitetaan uudelle kategorialle haluttu nimi ja painetaan Valmis-painiketta. Painike lähettää tiedon uudesta kategoriasta POST-metodilla. POST-metodin lähettämät tiedot otetaan vastaan PHP-koodissa ja lähetetään tietokantaan. Tietokanta lisää itse id-numeron uudelle kategorialle. Tämä tapahtuu MySQL:stä löytyvällä AUTO_INCREMENT -ominaisuudella, joka tuottaa aina uuden uniikin numeron, joka on isompi kuin isoin sarakkeesta jo löytyvä numero. Lisäyksen jälkeen uusi kategorian on nähtävillä kategorialistauksessa.

6.7.4 Tuotteiden hallinta

Tuotteiden hallintasivulla ylläpitäjä näkee listauksen verkkokaupan tuotteista. Tällä sivulla ei kuitenkaan näytetä samaa kategorialistausta joka on jo nähtävillä verkkokaupan vasemmassa reunassa, vaan kaikki tuotteet on koottu yhteen listaan. Tästä listalta on tarkoitus muun muassa nähdä helposti poikkeavuuksia tuotteiden tiedoissa, jotta ne voidaan korjata helposti. Mikäli on tarkoitus muuttaa jonkin tietyn tuotteen tietoja, on tämän löytämistä varten lisätty myös haku. Tähän hakuun voi laittaa hakuehdoksi joko tuotteet nimen tai tuotenumeron (id-numero). Tuotenumeron perusteella ylläpitäjä voi helposti hakea oikean tuotteen. Tämän numeron voi löytää selaamalla verkkokaupan kategorioiden tuotteita aivan kuten asiakkaat ja kopioimalla sitten tuotenumeron tuotteen tiedoista.

Tuotelistauksesta tai hausta löytyneitä tuotteita pääsee muokkaamaan painamalla tuotenumeroa. Tällä sivulla näkyy kaikki tuotteen tiedot lomakkeessa, jossa niitä voi myös muokata. Tuotteilta löytyy tietokannan taulusta sarake ”myynnissä.” Tämän sarakkeen arvoa muuttamalla voi vaihtaa tuotteen näkyvyyttä verkkokaupan asiakkaille. Arvon ollessa luku 0 tuote ei ole näkyvillä ja sen ollessa luku 1 tuote on näkyvillä. Päivitä-painiketta painamalla tiedot päivitetään tietokantaan.

Kuten kategorioita, myös tuotteita voidaan lisätä vastaavasti hallintapaneelistä. Tämä tapahtuu painamalla ”Lisää uusi tuote” -painiketta, jolloin aukeaa uusi sivu. Tällä

sivulla olevaan lomakkeeseen ilmoitetaan tuotteelle haluttu nimi, hinta, kategoria, kuvaus ja varastosaldo. Lisää-painikkeella tiedot lähetetään tietokantaan. Oletuksena uusille tuotteille asetetaan ”myynnissä” sarakkeeseen arvoksi luku 1.

7 YHTEENVETO

7.1.1 Suunnittelun arviointi

Verkkokaupassa oli todella paljon eri asioita mitä olisi voinut vielä tehdä paremmaksi, eikä loppua olisi ollut näkyvissä ilman kunnollista suunnitelmaa projektille. Raja oli vedettävä johonkin sekä ajankäytössä, että ominaisuuksien toteuttamisessa. Tähän asiaan helpotusta toi hyvä aikataulus.

Aikataulutuksesta oli suuri hyöty projektin edetessä. Aikataulua seuraamalla pysyi mukana, paljonko aikaa oli vielä käytettävänä kunkin ominaisuuden toteuttamiseen ja miten laajaksi sitä kannatti alkaa rakentaa ennen kuin oli siirryttävä seuraavaan tehtävään. Työhön käytetty aika oli aikataulun mukaisesti noin yksi kuukausi, jonka aikana sitä tuli tehtyä jokaisena päivänä tyypillisen työpäivän verran. Kaiken kaikkiaan vaiheiden ajanjako osoittautui todella tärkeäksi, ettei yhden asian hiomiseen tulut käytettyä liikaa aikaa. Tärkeämpää on saada perustukset aluksi kuntoon ja vasta sen jälkeen tekemään lisäominaisuuksia.

7.1.2 Työn itsearviointi

Työ oli sopivan haastava ja vaati välillä tarvetta hyvälle ongelmanratkaisutaidoille. Ohjelmointiongelmille tyypilliseen tapaan pulmat kuitenkin ratkesivat hakukoneita käyttämällä ja dokumentaatioita lukemalla. Osa käytettävistä tekniikoista oli entuudestaan tuttuja, mutta myös uusia tulokkaita oli tarjolla. Vaikka käytinkin MySQL-ohjelmistoa nyt ensimmäistä kertaa kunnolla, oli sen käyttö silti luontevaa, sillä eroavaisuudet sen minulle entuudestaan tuttuun kilpailijaan olivat melko pienet. Enemmän uutta opittavaa tarjosi PHP-ohjelmointikieli, jonka dokumentaatiota tuli-

kin luettua aina ohjelmoinnin tiimassa. Työtä tehdessä oppimani uudet tekniikat ja entistä paremmiksi hiotut ohjelmointitaidot tulevat varmasti hyötykäyttöön myös tulevaisuudessa.

LÄHTEET

W3C. 2014. Open Web Platform Milestone Achieved with HTML5 Recommendation. Viitattu 12.2.2015. <http://www.w3.org/2014/10/html5-rec>

W3Techs. 2015. Historical yearly trends in the usage of server-side programming languages for websites. Viitattu 26.2.2015.
http://w3techs.com/technologies/history_overview/programming_language/ms/y

MySQL. 2015. MySQL Customers. Viitattu 27.2.2015.
<https://www.mysql.com/customers/>

PHP. 2013. PHP 5 ChangeLog. Viitattu 12.2.2015. <http://php.net/ChangeLog-5.php#5.5.0>