



Jaakko Hämeenaho & Mikko Koskela

WEB-SIVUSTO OULUN ALUEEN TAPAHTUMILLE

WEB-SIVUSTO OULUN ALUEEN TAPAHTUMILLE

Jaakko Hämeenaho
Mikko Koskela
Opinnäytetyö
Syksy 2013
Tietojenkäsittelyn koulutusohjelma
Oulun seudun ammattikorkeakoulu

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma

Tekijät: Jaakko Hämeenaho & Mikko Koskela
Opinnäytetyön nimi: Web-sivusto Oulun alueen tapahtumille
Työn ohjaaja: Pekka Ojala
Työn valmistumislukukausi ja -vuosi: Syksy 2013
Sivumäärä: 33

Tämän opinnäytetyön aiheena oli suunnitella, toteuttaa ja ottaa käyttöön web-sivusto Oulun alueen tapahtumille. Tässä raportissa perehdytään web-sivuston suunnitteluun ja web-sivuston toteutuksessa käytettyihin menetelmiin, sekä perehdytään automatisoidusti toimivaan tiedon keräämiseen RSS-syötteistä. Opinnäytetyöllä ei ole erillistä toimeksiantajaa. Tavoitteena oli luoda sivustosta helppokäyttöinen ja mahdollisimman automatisoidusti ylläpidettävä. Tarkoituksena oli ottaa valmis sivusto omaan käyttöön.

Tässä raportissa perehdytään tämän toiminnallisen opinnäytetyön toteutusvaiheisiin ja niissä käytettyihin tekniikoihin. Sivustolle toteutettiin kaikki sen tärkeimmät toiminnalliset ominaisuudet. Sivuston graafinen ilme toteutettiin erillisenä projektina.

Lopputuloksena sivusto julkaistiin Oulun alueen tapahtumille osoitteessa oulusa.fi. Tulevaisuudessa sivustoa on tarkoitus jatkokehittää luomalla sivustolle lisäominaisuuksia ja laajentaa sivustolla olevaa tapahtumatietokantaa.

Asiasanat: web-sivusto, portaali, PHP, RSS, Cron, JavaScript, SQL

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Business Information Systems

Authors: Jaakko Hämeenaho & Mikko Koskela
Title of thesis: Website for events in Oulu area
Supervisor: Pekka Ojala
Term and year when the thesis was submitted: Autumn 2013
Number of pages: 33

The aim of this thesis was to design and implement a website for events in Oulu area. This report discusses website designing, the methods used in the implementation, and collecting data automatically from RSS feeds. This thesis didn't have a client. The aim was to make the website easy to use to enable automatic maintenance, so it could be deployed by the authors.

More specifically this report describes implementation steps and the techniques used in this functional thesis. All the critical functions were implemented in the website. The graphic design for the website was carried out as a separate project.

As a result a website for events in Oulu area was published at oulusa.fi. In the future the intention is to further develop the website by creating additional features and expanding the event database.

Keywords: website, portal, PHP, RSS, Cron, JavaScript, SQL

SISÄLLYS

1 JOHDANTO	6
2 PORTAALIT	7
3 TEKNIIKAT JA MENETELMÄT	9
3.1 HTML	9
3.2 CSS	10
3.3 PHP	11
3.4 JavaScript	12
3.5 SQL	13
3.6 Cron	14
3.7 RSS	15
3.8 jQuery	15
4 SUUNNITTELU	17
4.1 Rakenteen ja toiminnallisuuden suunnittelu	17
4.2 Responsiivinen suunnittelu	18
4.3 Tietokannan suunnittelu	19
4.4 Sisällön suunnittelu	19
4.5 Sivuston kohderyhmä	20
4.6 Tekijänoikeus	21
5 TOTEUTUS	22
5.1 Tavoitteet	22
5.2 Projektin suunnittelu ja aikataulutus	23
5.3 Tekninen toteutus ja testaus	25
5.4 Graafisen suunnitelman käyttöönotto	27
5.6 Sivuston julkaisu ja ylläpito	28
6 POHDINTA	29
LÄHTEET	31

1 JOHDANTO

Tämän opinnäytetyön aiheena on suunnitella ja toteuttaa web-sivusto, jolla käyttäjät voivat selata Oulun alueen tapahtumia. Halusimme toteuttaa tämän sivuston omiin tarpeisiimme, joten projektille ei ole ulkopuolista toimeksiantajaa. Tässä raportissa kerrotaan niistä menetelmistä, joilla on toteutettu opinnäytetyön toiminnallinen osuus.

Sivustolle luodaan oma tietokanta web-palvelimelle, johon kerätään automatisoidusti tapahtumatiedot muilta www-sivustoilta. Tietokannan tapahtumalähteiksi valitaan vähintään yksi RSS-syöte. Sivustoja, joilta tapahtumatietoja tietokantaan kerätään ovat 45 Special ja Oulun kaupunki. Tavoitteenamme on tulevaisuudessa laajentaa tietokantaa siten, että mukaan saadaan kaikki Oulun alueen tapahtumat. Sivuston tarkoituksena on tarjota uusi markkinointikanava Oulussa tapahtuville tapahtumille.

Sivustolla käyttäjän on mahdollista valita haluamansa näkymä tapahtumille muun muassa valitun päivän, kategorian tai järjestäjän mukaan. Käyttäjän on myös mahdollista äänestää tapahtumia sekä selata tapahtumia muiden antamien äänien mukaan.

Tässä opinnäytetyössä tutkitaan web-sivuston toteuttamisen kannalta oleellisimpia teknisiä menetelmiä, joita ovat HTML, JavaScript, CSS, PHP, SQL, Cron ja RSS. Opinnäytetyössä käydään läpi myös web-sivuston suunnittelu- ja toteutusvaiheen keskeisimmät asiat.

2 PORTAALIT

Sivuston toiminta perustuu siihen, että se hakee muilta web-sivustoilta tietoa joka on kerätty sivustolle yhdeksi kokonaisuudeksi. Tällaista palvelua voidaan kutsua verkkoportaaliksi, sillä palvelu tarjoaa omien toimintojensa lisäksi pääsyn useisiin muihin yhdenmukaisiin verkkopalveluihin esimerkiksi linkillä.

Verkkoportaali on verkkopalvelu, joka auttavat hahmottamaan internetissä saatavilla olevaa tietoa. Alunperin verkkoportaaleilla viitattiin tunnettuihin internetin hakusivustoihin, jotka mahdollistivat käyttäjille pääsyn informaatioon verkossa. Monet mieltävät nykyään tunnetuimmat verkkoportaali hakukoneiksi, vaikka ne tarjoavat paljon muutakin ominaisuuksia kuin tiedon haun internetistä. Vertaillen verkkoportaaleja hakukoneisiin, tarjoavat verkkoportaali rakenteellisemmän ja helppokäyttöisemmän käyttöliittymän. Tiedon hakeminen verkkoportaaleista, joihin tieto on syötetty valituilta sivustoilta on yleensä tehokkaampi tapa, kuin pelkällä hakusanalla hakeminen koko internetistä. Samalla kun julkiset verkkoportaali ovat laajentuneet, ovat portaalitekniikoiden markkinat keskittyneet yhä enemmän yrityspuolelle. (Tatum 2013, hakupäivä 13.11.2013; Winkler, hakupäivä 13.11.2013.)

Verkkoportaali on verkkopalvelu joka tarjoaa tietoa useasta lähteestä. Ideana on, että verkkoportaali toimii aloituspisteenä käyttäjälle, joka hakee internetistä tietyn tyyppistä tietoa. Lähes kaikkien verkkoportaalien toiminta perustuu jonkin tyyppisen hakukoneen ympärille. Tunnetuimmat verkkoportaali sisältävät muun muassa kategorioittain tiedot suosituilta sivustoilta, uutisista, säästä, viihteestä ja talouskatsauksista. (Lister 2013, hakupäivä 13.11.2013.)

Tällaisia verkkoportaaleja on useita, joista hyvänä esimerkkinä toimii suomalainen ampparit.com. Sivustolle kerätään automaattisesti suomalaisten web-sivustojen uutisia useista lähteistä. Uutislinkistä käyttäjä pääsee siirtymään uutisen alkuperäiseen lähteeseen. Näin ollen portaalit toimii aloituspisteenä

käyttäjälle, joka haluaa seurata uutisia kootusti useammalta sivustolta. Työn tavoitteena on toteuttaa samankaltainen verkkoportaali Oulun alueen tapahtumille.

3 TEKNIIKAT JA MENETELMÄT

Verkkoportaalin luonnissa tarvitaan useita tekniikoita ja työkaluja. Luvussa käydään läpi web-sivuston luonnissa käytettävät tekniikat, kuten HTML, CSS, JavaScript, PHP sekä tutkitaan SQL-tietokannan rakennetta ja sen toiminnallisuutta. Lisäksi automatisoidulle verkkoportaalille tärkeät Cron ja RSS esitellään tässä kappaleessa.

3.1 HTML

HTML on lyhenne sanoista Hypertext Markup Language, mikä tarkoittaa suomennettuna hypertekstin merkkäuskieltä. Se on tärkein merkintäkieli web-sivustojen luonnissa ja tietojen esittämisessä web-selaimessa. Sen kehittäminen on aloitettu 1990-luvun alussa ja sen viimeisin julkaisuversio on vuonna 2012 julkaistu HTML5. (W3C 2011, hakupäivä 9.10.2013.)

HTML-dokumentti on tekstimuotoinen, mutta siinä voidaan viitata osoitteilla esimerkiksi kuviin, äänitallenteisiin ja animaatiota tuottaviin ohjelmiin. HTML-koodi rakentuu sisäkkäisistä ja peräkkäisistä elementeistä, joita ovat esimerkiksi linkit, erilaiset otsikot tai sivustolle tuleva kuva (kuvio 1). Elementit merkataan HTML-kielessä kulmasulkeilla merkityin tunnistein eli tagein. Elementeillä on tavallisesti aloitus- sekä lopetustunnisteet, joiden väliin kirjoitetaan elementin sisältö. Esimerkiksi halutun otsikon sisältö kirjoitetaan tunnisteiden `<h1>` ja `</h1>` väliin. Elementit voivat sisältää attribuutteja eli tarkentimia, jotka merkitään elementin kulmasulkeiden sisään kuten esimerkiksi `<h1 id="otsikko">`. (w3c 2011, hakupäivä 10.10.2013.)

```

<body>
<title>Tapahtumat</title>

<div id="tausta">

    <div id="ylätunniste">
        <a href="mobi/indexmobi.html">Mobiili</a>
    </div>

```

KUVIO 1. HTML kieltä

HTML-dokumentteja on mahdollista luoda millä tahansa tekstinkäsittelyohjelmalla. Lisäksi on olemassa useita sivujen tekemistä auttavia HTML-editoreja. Näitä editoreita kutsutaan myös wysiwyg-ohjelmiksi (What You See Is What You Get), joiden käyttö muistuttaa osittain tekstinkäsittelyohjelman käyttöä. Riskinä HTML-editoreita käytettäessä on se, että sivuston koodi ei pysy yhtä selkeänä kuin HTML-kielellä ohjelmoidessa. Toinen olennainen riski HTML-editoreita käytettäessä on se, että niillä luotu web-sivusto ei välttämättä toimi kaikissa selaimissa. (Korpela & Linjama 2004, 34-35.)

3.2 CSS

CSS eli Cascading Style Sheets on tekniikka, joka määrittää tyyliohjeet rakenteellisiin dokumentteihin kuten HTML-tiedostoon. CSS on kehitetty vuonna 1997 HTML-dokumenttien tueksi, jotta sivuston sisällöllistä ja graafista puolta olisi helpompi hallita. CSS-tiedostoon luodut ohjeet määrittävät viitatus dokumentin säännöt kuten asettelun, fontin ja värimaailman. CSS-tekniikan käyttö alkoi saavuttaa suosiota vuonna 2000, kun selaimet alkoivat tukea enemmän CSS-tiedostoihin annettuja ohjeita. Uusin CSS julkaisuversio on CSS3, jota suurin osa selaimista nykyään tukee. (Kyrnin, hakupäivä 10.10.2013.)

CSS-kirjoitussääntö muodostuu kahdesta osasta: valitsimesta ja määrittelyosasta. Valitsimessa määritetään minkä elementin ulkoasua halutaan määrittää. Valitsimet voivat olla koko sivuston kaikkia elementtejä, tietyn

elementin kaikkia esiintymiä, tietyn ID-attribuutin tai luokka-attribuutin elementtejä määrittäviä. Määrittelyosa koostuu ominaisuudesta ja arvosta. Määrittelyosa tulee valitsimen perään kaarisulkuihin, jonka sisällä ominaisuus ja arvo erotetaan kaksoispisteellä (kuvio 2). (Haverinen 2011, hakupäivä 10.10.2013.)

```
a {  
  text-decoration: bold;  
  color:red;  
}  
  
div#ylätunniste {  
  height: 100px;  
  width: 100%;  
  float:left;  
  background-color:yellow;  
  color: black;  
}
```

KUVIO 2. Tyylimäärittelyä CSS:llä

CSS-tiedosto voidaan yhdistää HTML-dokumenttiin sen head-osiossa link-elementillä. Link-elementti sopii hyvin käytettäväksi tilanteessa, jossa useampi dokumentti käyttää samaa tyylitiedostoa. Tällöin koko sivuston ulkoasu voidaan määrittää yhdessä CSS-dokumentissa ja ulkoasun hallinta helpottuu huomattavasti. (Saarikumpu 2013, hakupäivä 10.10.2013.)

3.3 PHP

PHP on avoimen lähdekoodin komentosarjakieli, jolla voidaan luoda dynaamisesti toimivia web-sivustoja. PHP-kielen syntaksi on helppo oppia ja se muistuttaa Perl ja C-ohjelmointikieliä. PHP:tä voidaan käyttää useilla eri alustoilla ja käyttöjärjestelmillä, mutta se on suunnattu erityisesti web-sovelluskehitykseen ja web-sivustojen toiminnallisuuden luomiseen. PHP tukee useimpia tavallisesti käytettyjä tietokantoja, minkä takia se on usein tärkeässä roolissa haettaessa tietoja tietokannasta web-sivustoille. (PHP Group 2013a,b, hakupäivä 11.10.2013.)

HTML-sivustolla PHP-koodi kirjoitetaan tagin aloitusosan <?php ja lopetusosan ?> väliin (kuvio 3). Dokumentit nimetään tavallisesti .php-päätteisiksi, jotta palvelin tunnistaa PHP-koodin sivulta. PHP-koodin sisältö tulkitaan HTML-kieleksi, joten PHP-koodia ei näy erikseen. (Nixon 2009, 34.)

```
]function teeOhjaus($variables){
    $ohjaus = "?";
    ] if (isset($variables["date"])){
        ..... $ohjaus .= "date=" . $variables["date"] . "&";
    }
    ] if (isset($variables["page"])){
        ..... $ohjaus .= "page=" . $variables["page"] . "&";
    }
    ] if (isset($variables["run"])){
        ..... $ohjaus .= "run=" . $variables["run"] . "&";
    }
    ] if (isset($variables["category"])){
        ..... $ohjaus .= "category=" . $variables["category"];
    }
    ] if (substr($ohjaus, -5) == "&"){
        ..... $ohjaus = substr($ohjaus, 0, -5);
    }
    return $ohjaus;
}]
```

KUVIO 3. PHP-kieltä

3.4 JavaScript

JavaScript on oliopohjainen komentosarjakieli, jolla PHP:n tapaan luodaan web-sivustolle dynaamista toiminnallisuutta ja interaktiivisuutta, jotka vastaavat käyttäjän tekemiin toimintoihin kuten hiiren napsautuksiin ja syöttötietojen kirjoittamiseen. Jos JavaScript-koodi on liitetty HTML-dokumenttiin, selain suorittaa tämän koodin samalla, kun se näyttää kyseisen HTML-dokumentin. Yksi syy mihin JavaScriptiä käytetään, on HTML-dokumenttien lomakkeiden yksinkertaisuus. HTML-dokumentissa ei ole mitään tapaa antaa käyttäjälle välitöntä palautetta, jos hän kirjoittaa lomakkeeseen tiedon väärässä muodossa. Jos kenttään vaaditaan numerollinen arvo ja käyttäjä kirjoittaa kenttään "sata", virhe ilmenee vasta, kun lomake on lähetetty palvelimen käsiteltäväksi. JavaScriptillä voidaan ilmoittaa käyttäjälle heti, jos kentän arvo on virheellinen,

jolloin käyttäjän on helpompi hahmottaa missä hän teki virheen. (Korpela 2005, 294.)

Toisin kuin HTML ja CSS, on JavaScript ohjelmointikieli, jota käytetään yleensä selainskriptien tekemiseen. Selainskriptien tekemiseen perustuvia ohjelmointikieliä on useita, mutta JavaScript on niistä yleisimmäin käytössä oleva, sillä suurin osa selaimista tukee sitä. JavaScript on monipuolinen ohjelmointikieli ja hyvin yksinkertaisillakin JavaScript-ohjelmilla pystyy tekemään monenlaisia toiminnallisuuksia web-sivustoille, sillä kielessä on etenkin selainkäyttöön tarkoitettuja erikoispiirteitä. (Korpela 2005, 295.)

```
<script>
    $(function() {
        $("#datepicker").datepicker({
            nextText: "Seuraava",
            prevText: "Edellinen",
            firstDay: 1,
            dayNames: [ "Sunnuntai", "Maanantai", "Tiistai", "Keskiviikko", "Torstai", "Perjantai", "Lauantai" ],
            dayNamesMin: [ "Su", "Ma", "Ti", "Ke", "To", "Pe", "La" ],
            monthNames: [ "Tammikuu", "Helmikuu", "Maaliskuu", "Huhtikuu", "Toukokuu", "Kesäkuu", "Heinäkuu", "Elo" ],
            dateFormat: 'yy-mm-dd',
            onSelect: function(dateText, inst) {
                window.location.href=window.location.href+"?date=" + dateText;
            }
        });
        var date = "<?php echo $date; ?>";
        var year = date.substr(0,4);
        var month = parseInt(date.substr(5,2));
        month = month - 1;
        var day = date.substr(8,2);
        $('#datepicker').datepicker('setDate', new Date(year, month, day));
    });
</script>
```

KUVIO 4. JavaScript-koodia

3.5 SQL

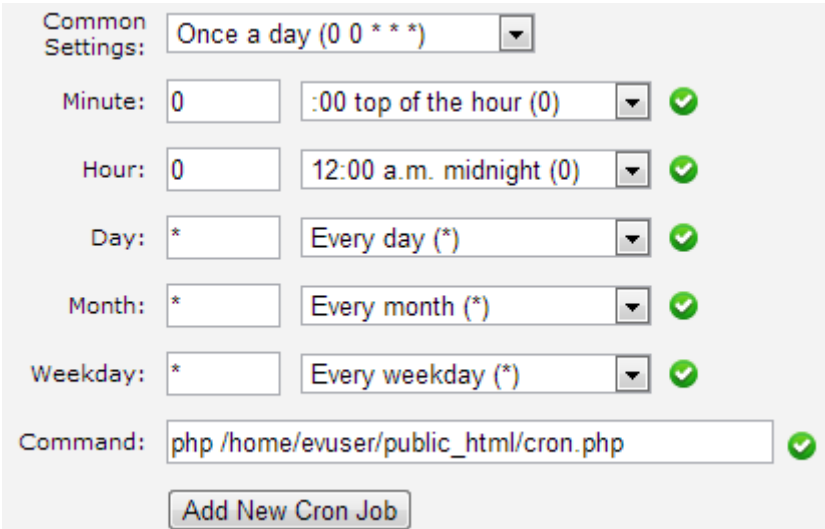
SQL on lyhenne sanoista Structured Query Language. SQL on tietokannan käsittelyyn standardoitu kyselykieli, jolla voidaan hakea tietoja, tehdä kyselyjä, muutoksia ja lisäyksiä käytännössä kaikkiin relaatiotietokantoihin. SQL on nykyään suosituin tietokannan hallintajärjestelmä, joka vaikuttaa useilla maailman suosituimmilla web-sivustoilla. Sen suosio perustuu sen toimivuuteen, nopeuteen ja ilmaiseen saatavuuteen. (Nixon 2009, 157.)

Yleisimmät SQL-kielellä toteutettavat toiminnot ovat tietokannan rakenteen määrittely ja muuttaminen, kyselyt tietokantaan, laskenta tietokannan datalla, tietojen lisääminen, muuttaminen ja poistaminen sekä käyttöoikeuksien määrittely. SQL sisältää peruskyselyt, joilla voidaan tehdä haku (SELECT), lisäys (INSERT) ja poistotoimenpiteet (DELETE). (Huotari 2013, hakupäivä 11.11.2013.)

3.6 Cron

Cron-tehtävä on Linux-palvelimen aikataulutettava komento skriptien ja komentojen ajamiseen. Sillä voidaan suorittaa toistuvia tehtäviä automaattisesti. Tyypillisesti cron-tehtävänä ajettavia skriptejä käytetään tiedostojen tai tietokantojen päivittämiseen.

Cronilla ajettavia tehtäviä voi myös käyttää muihin tehtäviin, kuten vaikka lähettämään sähköposti-ilmoituksia tai tuhoamaan väliaikaiset tiedostot viikon välein. Cron-tehtävä voi olla hyvin tehokas työkalu, koska sen voi laittaa suoriutumaan minuutin, tunnin, päivän, viikonpäivän tai kuukauden tarkkuudella (kuvio 5). (HostGator, hakupäivä 25.10.2013.)



Common Settings: Once a day (0 0 * * *)

Minute: 0 :00 top of the hour (0) ✓

Hour: 0 12:00 a.m. midnight (0) ✓

Day: * Every day (*) ✓

Month: * Every month (*) ✓

Weekday: * Every weekday (*) ✓

Command: php /home/evuser/public_html/cron.php ✓

Add New Cron Job

KUVIO 5. Cron-tehtävän lisääminen cPanelissa

3.7 RSS

RSS on lyhenne sanoista Really Simple Syndication. RSS on vaivaa säästävää työkalu, joka mahdollistaa mielenkiintoisten informaatiolähteiden seuraamisen, kuten blogin, podcastin, videocastin tai minkä tahansa web-sivuston, jolla on RSS-syötteet käytössä. RSS-syötteiden käyttö korostuu, kun käytetään useita syötteitä yhtäaikaaisesti. Tällöin voidaan seurata useaa informaatiolähdettä minimaalisella vaivalla. RSS-syötteiden ensisijainen arvo on, että se tuo informaation käyttäjälle ilman useilla sivustoilla käymistä. RSS-syötteitä voi käyttää esimerkiksi jakamaan informaatiota tapahtumista. (Shapiro 2006, hakupäivä 25.10.2013.)

Nimi RSS on laaja määritelmä formaatille, jolla on useita eri versiota kahden rinnakkaisen formaatin alla. Alkuperäinen RSS versio 0.90 oli Netscapen suunnittelema formaatti, jolla pystyi rakentamaan sivustoja aina uutisotsikkoportaaleista valtamedian uutissivustoihin. Sen uskottiin olevan liian monimutkainen, joten luotiin yksinkertaisempi versio 0.91. Netscape menetti mielenkiinnon portaalien luomiseen ja täten lopetti RSS:n kehittämisen. Kuitenkin eri kehittäjä, UserLand Software oli kiinnostunut jatkokehityksestä ja jatkoi siitä mihin Netscape jäi. Sillä aikaa kolmas, ei-kaupallinen ryhmä, suunnitteli uuden formaatin 1.0-version, joka perustui 0.90-versioon. UserLand ei ollut mukana kehittämässä 1.0-versiota eikä ollut siihen tyytyväinen, joten se jatkoi kehittämällä omat versionsa 0.92, 0.93, 0.94 ja lopulta nykyään laajimmin käytössä olevan version 2.0. (Pilgrim 2002, hakupäivä 1.11.2013.)

3.8 jQuery

jQuery ei ole ohjelmointikieli vaan se on avoimesti käytettävissä olevaa valmiiksi kirjoitettua JavaScript-koodia. Se on nopea ja tiivis JavaScript-kirjasto, joka yksinkertaistaa HTML-dokumenttien läpikäymisen, tapahtumakäsittelyn, animoinnin ja Ajax-vuorovaikuttamisen. jQuery:n käyttö mahdollistaa keskeisten käyttöliittymätoiminnallisuuden kirjoittamisen vähäisillä määrillä koodia. (Narayan 2011, hakupäivä 13.11.2013.)

jQuery parantaa sovelluksen suorituskykyä ja selainyhteensopivuutta. Se vähentää koodin kirjoittamista käyttöliittymää toteutettaessa ja se on laajennettavissa luomaan haluttua käytöstä. Lisäksi sen käyttöön ei tarvitse opetella uudenlaista syntaksia, sillä JavaScriptin osaaminen riittää. (Narayan 2011, hakupäivä 13.11.2013.)

4 SUUNNITTELU

Tässä luvussa käydään läpi web-sivuston ulkoasun, sisällön ja rakenteen suunnittelun kannalta tärkeimmät asiat. Lisäksi kerrotaan miten web-sivustoilla tulisi ottaa huomioon kohderyhmän rajaaminen ja tekijänoikeuskysymykset.

4.1 Rakenteen ja toiminnallisuuden suunnittelu

Sivuston rakenne on hierarkkinen siten, että pääsivulta voi siirtyä alemman tason sivulle, jossa voi olla linkkejä "alaspäin" ja niin edelleen. Rakenteen ei tarvitse välttämättä olla tällainen, vaan sivut voivat olla linkitetty myös ristiin. Hierarkkisen rakenteen etuna on se, että sivuston oleellisimmat asiat tulevat käyttäjän nähtäväksi ensimmäisenä. Oleellista pääsivulla on, että se antaa kuvan koko sivuston laajuudesta sekä yleistä tietoa sivustosta. (Korpela & Linjama 2005, 134; Korpela 2009a, hakupäivä 15.10.2013.)

Sivuston suunnittelu kannattaa tehdä hierarkkisesti sivuston tärkeimmistä asioista yksityiskohtiin edeten, mutta toteutusvaiheessa toimitaan päinvastoin siten, että ylemmän tason sivulta alaspäin luodaan linkki vasta, kun se johtaa johonkin eli alasivu on tehty ensin. Alasivuilla tulisi olla sivuston mukainen looginen viittaus ylöspäin sivustohierarkiassa. Pienessä sivustossa pelkästään linkki pääsivulle on riittävä. Mitä alemmas sivustohierarkiassa mennään, sitä enemmän varsinaisen sisällön tulisi korostua. (Korpela & Linjama 2005, 135; Korpela 2009a, hakupäivä 15.10.2013.)

Web-sivustoilla tulisi pitää rakenne mahdollisimman samanlaisena kaikilla sivuilla. Tämä on tärkeää varsinkin keskeisimmillä sivuilla, mutta se on eduksi myös alasivuilla. Näin käyttäjä löytää aina samat perustiedot samoista paikoista. Yksinkertainen tapa luoda sivuista yhtenäisiä ja rakenteeltaan samanlaisia on luoda sivurunko. Sivurunko helpottaa sisällön tekemistä, koska sitä voidaan

käyttää valmiina pohjana kaikille sivuston sivuille. (Korpela 2009a,b, hakupäivä 15.10.2013.)

Sivuston käytettävyyden helpottamiseksi tulee suunnitteluvaiheessa miettiä sivuston navigaatiota. Selkeä navigaatio on jatkuva ja se auttaa käyttäjää hahmottamaan sivuston rakennetta. Sivuston päänavigaatio tulee pääsivulle useimmiten vaaka- tai pystypalkkina, ja sen tulisi olla selvästi erottuva. Päänavigaation linkkien tulisi kuvata selkeästi ja lyhyesti mihin sivuston alisivuille ne johtavat. (Järvenpää J., hakupäivä 15.10.2013; Jyväskylän Yliopisto, hakupäivä 15.10.2013.)

4.2 Responsiivinen suunnittelu

Responsiivisella suunnittelulla (responsive design) tarkoitetaan sellaista sivuston asettelua, että sivuston rakenne ja asettelu suhteutetaan näytön kokoon. Responsiivista suunnittelua tarvitaan erityisesti silloin kun oletuksena on, että sivustoa tullaan käyttämään eri kokoisilla näyttöpäätteillä. Responsiivisesti toimiva sivusto voidaan toteuttaa nykyisin pelkällä CSS:llä. (Korpela J. K. 2012, hakupäivä 11.11.2013.)

Responsiivisessa suunnittelussa huomioidaan laitteen käyttötapaa eli selaimen käytettävissä olevan alueen kokoa. Nykyainen tapa toteuttaa responsiivisuus, on käyttää CSS3:n mediarajoituksia (media queries). Esimerkiksi CSS-koodi tilanteisiin, joissa näytön leveys on enintään 640 pikseliä toteutetaan, kuten kuviossa 6.

```
@media only screen and (max-width: 640px) {  
  /* CSS-määritelmät */  
}
```

KUVIO 6. Responsiivisuuden määrittäminen CSS:llä

Käytännössä CSS-määritykset rajataan muutamalle erikokoiselle ikkunalle tai laitteelle, jotta sivusto skaalautuisi oikein kaikilla näyttöpäätteillä. Yleensä on kannattavaa kirjoittaa näyttökohtaiset tyylimääräykset omiin CSS-tiedostoihin, jolloin mediarajoitus esitetään link-elementissä. (Korpela J. K. 2012, hakupäivä 11.11.2013.)

4.3 Tietokannan suunnittelu

On erittäin tärkeä suunnitella tietokanta ennen kuin alkaa rakentamaan sitä, koska muuten tietokannan rakennetta joutuu melko varmasti muuttamaan. Suunnittelu kannattaa aloittaa kirjoittamalla ylös mitä kaikkia tietoja tietokannasta käyttäjät tulevat hakemaan ja miten ne halutaan esittää. Näin saadaan ratkaistua miten tietokannan taulukot tulee yhdistää toisiinsa. Hyvä tavoite tietokannan suunnittelussa on, että tietoa on helppo lisätä, hakea ja muuttaa. (Nixon, 2009, 201-202.)

Tietokannan suunnitteluvaiheessa päätetään tietokannan taulut, niiden väliset suhteet, taulujen kentät ja niihin tulevat tiedot. Kun tietokantaan lisätään uusia tietoja, sen rakenne ei saa enää muuttua. Ainoa muuttuva asia tietokannassa on taulujen rivit. Taulut ja niiden kentät ovat pysyviä. Sama tieto tulisi esiintyä tietokannassa vain yhdessä paikassa ja yksi tietokannan kenttä saa sisältää vain yhden tiedon. Useimmat tietokannan tauluista ovat yhteydessä toisiinsa. Taulujen välisiä yhteyksiä ovat yksi-moneen ja monta-moneen. Taulujen väliset suhteet suunnitellaan ennen tietokannan rakentamista. (Laaksonen, A. 2009, hakupäivä 25.10.2013.)

4.4 Sisällön suunnittelu

Sisällön suunnittelu tulisi aloittaa suunnittelemalla aluksi yksinkertainen ja nopeasti toteutettava sivusto. On helpompaa luoda ensin suppeampi, mutta siisti sivusto ja alkaa laajentamaan sitä, kuin luoda tyhjästä massiivinen sivusto. Sisältöä suunnitellessa ideoiden ylöskirjaaminen on kannattavaa, sillä kun on

luotu hyvä pohja, voidaan valita ideoista toteutettaviksi ne, jotka yhä vaikuttavat tärkeiltä ja tekemisen arvoisilta. Sivustojen ylläpitämisen työmäärää on vaikea arvioida suunnitteluvaiheessa, minkä johdosta uuden sisällön lisäämisen tarve voidaan arvioida paremmin vasta ensimmäisen sivustonjulkaisun jälkeen. (Korpela & Linjama, 2005, 56.)

Suunnitteluvaiheessa tulisi miettiä sivuston luontia, kehittämistä ja hoitoa prosessina, joka jatkuu sivuston julkaisemisen jälkeenkin. Tämän takia olisi jo alkuvaiheessa hyödyllistä tehdä pääpiirteittäinen kunnossapitosuunnitelma, jotta päivittäminen sujuisi ongelmitta. Kunnossapitosuunnitelmaan voi kuulua esimerkiksi; kaikkien sivujen läpisilmäily tietyn ajanjakson välein, sisällön päivittäminen, käyttäjäpalautteiden huomioiminen ja niissä ilmaantuneiden ongelmien korjaaminen välittömästi. (Korpela & Linjama, 2005, 53-54.)

4.5 Sivuston kohderyhmä

Web-sivustoa luodessa perusasetelma on se, että kuka tahansa voi käydä sivustolla, mutta kannattaa silti suunnitteluvaiheessa miettiä millaisille käyttäjille sivusto on ensisijaisesti tarkoitettu. Sivuston pääkohderyhmän rajausta vaikuttaa sivuston esitystapoihin, kuten kieleen, grafiikkaan ja viestintätapoihin. Sivustolla voi olla useita kohderyhmiä, jotka voivat olla keskeisessä asemassa. (Korpela & Linjama, 2005, 49.)

Yleisesti viestintä on helpompaa suppeammalle kohderyhmälle, mutta web-sivustoa luodessa ei yleensä ole syytä tehdä suuria rajoituksia vain kohderyhmän rajaamiseksi, sillä sivustojen tarkoituksena usein on tavoitella mahdollisimman suurta kävijäjoukkoa. Sivustolla tulisi tavoitella yleisyyttä, yleisesti ymmärrettävyyttä ja kiinnostavuutta, jotta sivusto pysyisi pääkohderyhmän lisäksi kaikille ymmärrettävänä ja helposti lähestyttävänä. Web-sivustolla kohderyhmän rajaamisessa kyse on erityisesti siitä, millaiset käyttäjät pyritään erityisesti ottamaan sivustolla huomioon, rajaamatta ketään kuitenkaan kohderyhmän ulkopuolelle. (Korpela & Linjama, 2005, 50.)

4.6 Tekijänoikeus

Tekijänoikeus on yleisimpiä asioita, joita yksityiset ihmiset loukkaavat laatiessaan web-sivustoja. Internetissä julkaistu aineisto muodostaa tyypillisesti teoksen silloin, kun se saatetaan yleisön saataviin. Tällöin web-julkaisu saa saman tekijänoikeussuojan kuin muutkin teokset. Tekijänoikeuslain peruseriaatteen mukaisesti teoksen saattaminen yleisön saataville on tekijän yksinomainen oikeus, eikä teosta saa kukaan muu julkaista ilman alkuperäisen tekijän lupaa. (Korpela & Linjama, 2005, 65-66.)

Tekijänoikeus koskee lain mukaan teosta ”muuttumattomana tai muutettuna, käännöksenä tai muunnelmana, toisessa kirjallisuus- tai taidelajissa taikka toista tekotapaa käyttäen”. Toisella web-sivustolla olevaa sisältöä ei yleensä saa käyttää omalla sivustolla ilman tekijän lupaa. Käyttämiseksi tulkitaan myös kuvan upottaminen sivustolle, vaikka img-rakenteessa viitattaisiin kuvan alkuperäiseen sijaintipaikkaan. Toiselle sivustolle viittaaminen asianmukaisesti linkillä on sallittua. Linkittämisessä tulee muistaa, että toisen sivustoa ei saa esittää niin sanotusti oman sivuston kehyksessä. (Korpela & Linjama, 2005, 66.)

Yleisesti jaossa olevaa tapahtumatietoa saa jakaa omassa verkkoportaalissa siinä tapauksessa, että kyseessä ei ole yksityistapahtuma. Muiden tapahtumien kuvien tai logojen esittäminen omalla sivustolla voi rikkoa tekijänoikeuslakia, jos kuva esitetään ilman alkuperäisen esittäjän lupaa. Kuvan tai minkä tahansa teoksen tekijä voi esittää luvan myöntämiselle mitä ehtoja haluaa, joista yleisin on rahallinen korvaus. Omassa verkkoportaalissa saa ilman tekijän suostumusta esittää toisella web-sivustoilla esiintyvistä tapahtumista nimen, ajan ja paikan. (Korpela & Linjama, 2005, 66.)

5 TOTEUTUS

Tässä luvussa käydään läpi sivustolle asetetut tavoitteet sekä toteutuksen vaiheet. Sivuston toteutus aloitettiin suunnittelulla ja aikataulun määrittelyllä. Luvussa perehdytään teknisessä toteutuksessa päädyttyihin ratkaisuihin ja vastaan tullessiin ongelmiin, sekä käydään läpi sivuston graafisen suunnitelman ratkaisuja. Luvussa tutustutaan myös sivustolla käytettyihin testausmenetelmiin sekä sivuston julkaisuun.

5.1 Tavoitteet

Opinnäytetyöprojektin tavoite oli kehittää Oulun alueen tapahtumille oma verkkoportaali, joka automaattisesti kerää kaikki Oulun alueen tapahtumat yhdelle sivustolle, josta käyttäjä voi selata niitä valitsemiensa asetusten mukaan. Sivuston tärkeimmät tavoitteet olivat automaattinen tietojen haku muilta sivustoilta RSS-syötteiden avulla sekä sivustolla näkyvien tapahtumien päivittyminen automaattisesti. Tavoitteena oli luoda Oulun alueen kattavin ja suosituin tapahtumaportaali.

Sivuston tuli olla mahdollisimman helppokäyttöinen kaikille sivustolla vieraileville sekä ulkoasun mahdollisimman selkeä ja kiinnostava. Sivuston näkymä tuli skaalautua näyttöpäätteelle sopivaksi. Sivuston ylläpitäminen tuli luoda mahdollisimman automatisoiduksi ja helppokäyttöiseksi. Sivuston julkaisun jälkeen sivustolle pyrittiin luomaan mahdollisimman suuri käyttäjäjoukko markkinoimalla sivustoa muun muassa sosiaalisessa mediassa.

Sivustolla käyttäjän on mahdollista valita erilaisia näkymiä tapahtuman ajan, kategorian ja tapahtuman suosion mukaan. Käyttäjällä on mahdollisuus äänestää ja kommentoida haluamaansa tapahtumaa, sekä päästä katsomaan lisätietoja linkin kautta tapahtuman alkuperäisellä sivulla. Lisäksi luodaan käyttäjille palautesivu.

5.2 Projektin suunnittelu ja aikataulutus

Koska tapahtumaportaalin luonti alkoi täysin tyhjältä pöydältä, täytyi pohtia mikä olisi paras tapa toteuttaa tällainen palvelu. Vaihtoehtoja olivat perinteinen web-sivusto, mobiilisovellus ja pääosin mobiililaitteille kohdennettu sivusto. Alkuperäisenä ajatuksena oli luoda palvelu mahdollisimman monella näistä tavoista, mutta aikataulun vuoksi päätettiin valita vain yksi tapa. Aluksi suunniteltiin sivusto mobiililaitteille responsiivista suunnittelua apuna käyttäen, josta kuitenkin myöhemmin luovuttiin ja alettiin kehittämään perinteistä web-sivustoa ilman responsiivisuutta.

Sivuston suunnittelu aloitettiin tutustumalla muihin kilpaileviin tapahtumasivustoihin, kuten ouluthisweek.net ja Oulun kaupungin tapahtumakalenteri. Tutustumalla myös muiden kaupunkien tapahtumakalentereihin, luotiin visio toteutettavasta sivustosta. Korpelan (2005) mukaan suunnitteluvaiheessa tulisi edetä hierarkkisesti ylemmältä tasolta pienempiin yksityiskohtiin ja kirjata ylös kaikki ideat, joista myöhemmin valittaisiin oleelliset. Suunnittelu aloitettiin valitsemalla ensin yksi sivusto, jolta tapahtumia haetaan. Seuraavaksi suunniteltiin sivustolle yksinkertainen käyttöliittymä ja asettelu sekä kirjattiin ylös ideoita, joita haluttiin sisällyttää sivustolle (kuvio 7).

Mobiili

Tähän filtitit, esim kuten rantapallo.fi

Kalenteri

Kategoriat

Haku

Pvm -

Pvm

Nimi

Järjestäjä: dropdownbok

HAE!

Top30

VIKKO /ETUSIVU

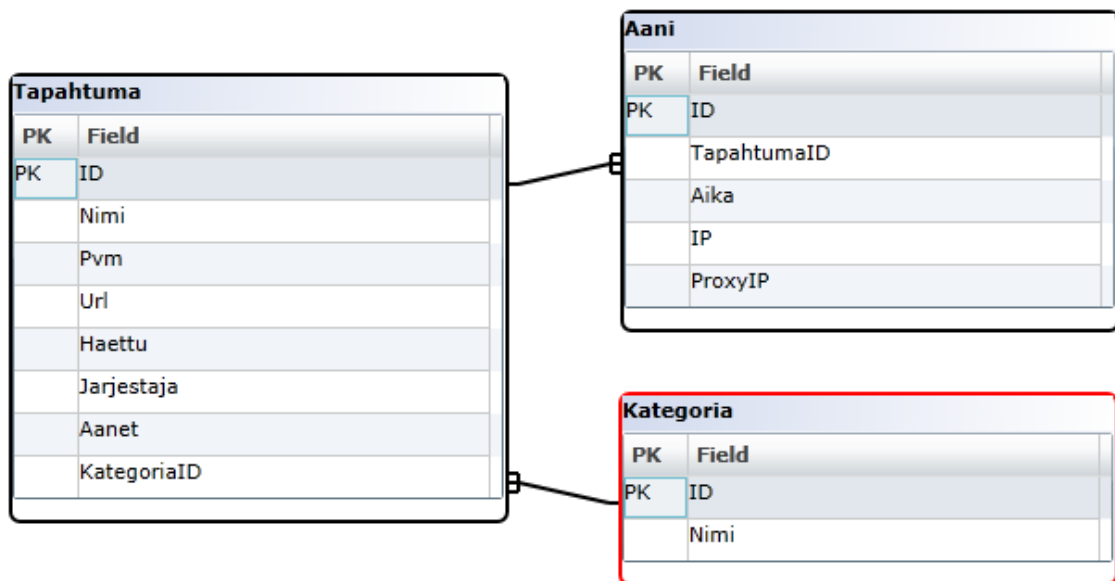
Suosituimmat

PALAUTE

Tapahtuman nimi	Aika	Järjestäjä	Suosittelut	Suosittele
Perjantai 20.09.2013				
Fajitas friday	16:00	45 Special	3	
Heikki kuula	22:00	45 Special	3	
Lauantai 21.09.2013				
45:n rokkidisko & studio 45	23:00	45 Special	2	
Sunnuntai 22.09.2013				
Sunnuntaijamit	22:00	45 Special	3	
Maanantai 23.09.2013				
Oulu open decks	22:00	45 Special	3	
Tiistai 24.09.2013				
Remakka open mic stand up club	20:30	45 Special	1	
Red cup	23:00	45 Special	1	
Keskiviikko 25.09.2013				
Children of boom	22:00	45 Special	1	
Collaboration 2000 presents mo's special & children of boom	22:00	45 Special	3	
Torstai 26.09.2013				
Antero lindgren	22:00	45 Special	1	
Perjantai 27.09.2013				
Fajitas friday	16:00	45 Special	1	

KUVIO 7. Sivuston suunnittelua

Kuten Korpela (2005) toteaa, tulee tietokannan rakenne ja sen taulut suunnitella valmiiksi ennen tietokannan luontia. Tietokantaa suunniteltaessa päätettiin, että tietokantaan tullaan tekemään taulut tapahtumille, kategorioille ja käyttäjien antamille äänille (kuvio 8).

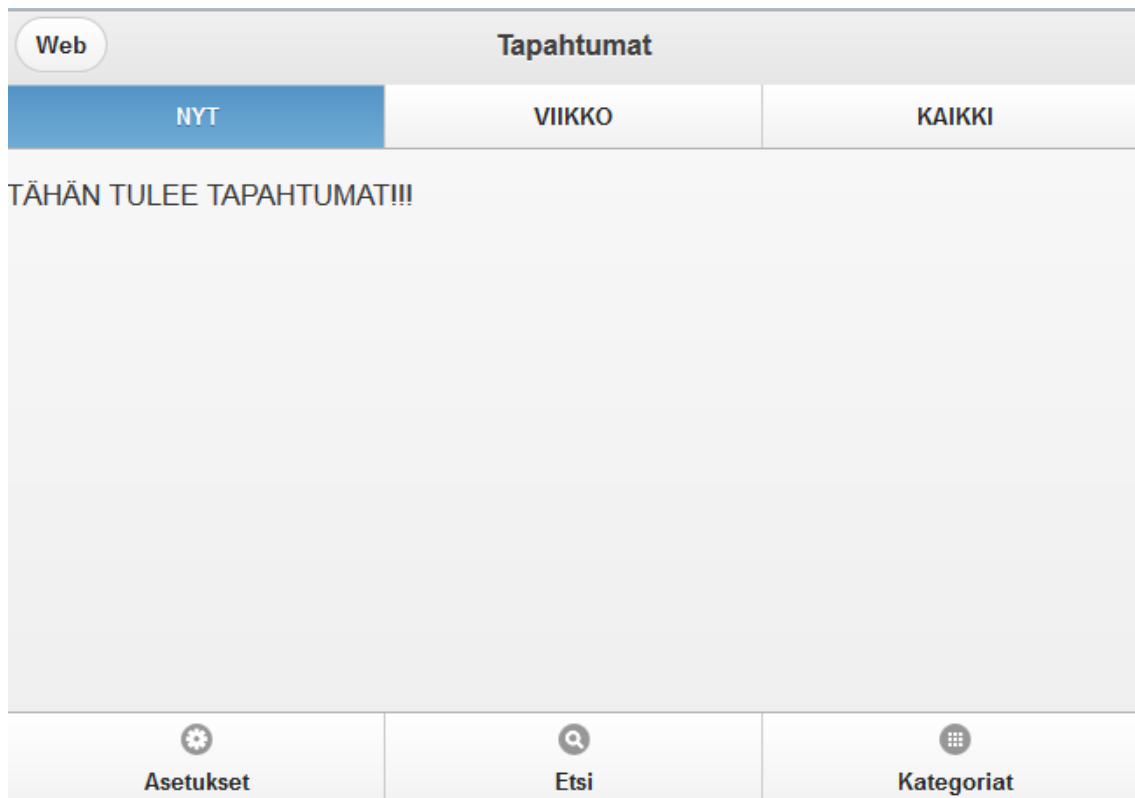


KUVIO 8. Tietokannan rakenne

Sivuston julkaisemisen takarajaksi asetettiin joulukuu 2013, joten sivuston suunnitteluun, toteutukseen ja testaukseen jäi aikaa noin neljä kuukautta. Suunnitteluvaiheessa päätettiin, että sivustolle luodaan erillisenä projektina graafinen suunnitelma, jotta aikataulussa pysyttäisiin paremmin.

5.3 Tekninen toteutus ja testaus

Sivuston luonti alkoi luomalla yksinkertainen pohja mobiilisivustolle responsiivisella suunnittelulla, joka toteutettiin käyttäen apuna jQuerya (kuvio 9). Samalla luotiin pohja perinteiselle web-sivustolle ilman responsiivisuutta ja jQuerya (kuvio 10). Jälkimmäisen ratkaisun todettiin olevan käyttäjäystävällisempi ja helpommin toteutettavissa. Idea mobiilisivustosta hylättiin ja kehitystä jatkettiin perinteisen web-sivuston pohjalta.



KUVIO 9. Mobiilikäyttöliittymä



KUVIO 10. Web-sivuston käyttöliittymä

Seuraava vaihe oli luoda tietokanta sekä toteuttaa RSS-syötteen tallentaminen tietokantaan. Tietokannan luonti oli helppoa, koska se oli suunniteltu etukäteen, joten se voitiin hyvin pitkälle luoda suunnitelman pohjalta. Ensimmäiseksi

tallennettavaksi RSS-syötteeksi valittiin 45 Specialin tapahtumasyöte, koska siinä tiedot löytyvät erittäin selkeästi ja riittävän kattavasti. Jokaiselle yksittäiselle RSS-syötteelle päätettiin tehdä oma PHP-tiedosto, joilla tiedot haetaan ja tallennetaan tietokantaan. Tietokannan automaattinen päivittyminen saavutettiin cron-tehtävillä, jotka ajavat tiedostoissa olevat skriptit ennalta määrätyin väliajoin. Myöhemmin lisättiin myös Oulun kaupungin RSS-syötteen tallentuminen tietokantaan.

Kun sivuston perusrakenne oli kunnossa, sekä tietokannassa oli esitettävää tietoa, alettiin sitä tuoda sivustolle. Tiedot tuodaan tietokannasta PHP:llä käyttäen SQL-kyselylauseita. Koska sivuston perustoiminnallisuus oli kunnossa, voitiin aloittaa erilaisten toimintojen toteutus sivustolle. Ensiksi toteutettiin tapahtumien äänestäminen ja suosituimpien tapahtumien katselu. Tämän jälkeen luotiin näkymät tapahtumien katseluun kategorioittain, päivämäärän sekä tekstihaun mukaan. Sivustolla oleva kalenteri on toteutettu JavaScriptillä, mutta muilta osin kaikki toiminnallisuus on tehty PHP:llä.

Koko sivusto on rakennettu yhteen kehykseen, sekä sivuston toiminnot ovat pääosin yhdessä tiedostossa. Tämä toteutustapa valittiin, koska tällöin kehittäminen on loogisempaa, eikä useiden tiedostojen päivittämisestä tarvitse erikseen huolehtia.

Sivustoa testattiin samalla, kun sitä kehitettiin ja viat korjattiin aina kun niitä löydettiin. Kun sivusto oli siinä kunnossa, että sitä voitiin käyttää, suoritettiin käyttäjätestaus. Lopuksi sivusto ajettiin kahden eri verkkosivustotestausohjelman läpi. Testit antoivat sivustosta hyvät tulokset, joten muutoksia ei tarvinnut vielä tässä vaiheessa tehdä.

5.4 Graafisen suunnitelman käyttöönotto

Sivuston graafinen suunnitelma tilattiin erillisenä projektina. Graafinen suunnitelma tulee sisältämään sivuston typografian, värimaailman, graafiset elementit ja asettelun, jotka tullaan ottamaan käyttöön.

Graafinen suunnitelma ei ehtinyt täysin valmistua sovittuun aikatauluun mennessä, joten sivustolle luotiin väliaikainen ulkoasu. Varsinainen suunnitelma tullaan ottamaan käyttöön, kun se on valmis.

5.6 Sivuston julkaisu ja ylläpito

Sivuston julkaisua varten täytyi sivustolle varata verkkotunnus sekä palvelin. Ideoita sopivasta verkkotunnuksesta kerättiin projektin aikana ylös. Verkkotunnuksen haluttiin olevan mahdollisimman hyvin sivustoa kuvaava ja sellainen, joka jäisi ihmisten mieleen. Ideoita verkkotunnuksiksi olivat muun muassa oulussa.fi, oulusa.fi, todo.fi ja tapahtumat.fi. Kun sivusto oltiin saatu toimivaksi, siirryttiin vertailemaan vapaita verkkotunnuksia viestintäviraston sivuille. Valituista verkkotunnuksista vapaana oli oulusa.fi, joten se varattiin.

Seuraavaksi vertailtiin eri web-hotelleja vertailusivustojen avulla. Vertailun tuloksena sivustolle sopivimmaksi web-hotelliksi osoittautui domainmaailma.fi, koska se oli edullinen ja siellä pystyi räätälöimään hinnan ja lisäpalvelut omien tarpeiden mukaiseksi. Domainmaailmasta valittiin melko suppea paketti, joka sisälsi 100 MB levytilaa, 1 GB kuukaudessa siirtokapasiteetin, yhden sähköpostitilin ja yhden tietokannan. Domainmaailman web-hotellin palveluita voi laajentaa tarvittaessa, joten näiden ominaisuuksien arvioitiin riittävän. Kun verkkotunnus sekä palvelin oli varattu, avattiin sivusto osoitteessa oulusa.fi.

Sivuston ylläpito luotiin mahdollisimman automatisoiduksi cron-tehtävien ja RSS-syötteiden avulla. Näin ollen sivuston ylläpito tulee sisältämään pääasiassa sivuston yleisen toiminnallisuuden ja linkkien toimivuuden tarkastelua. Sivuston ylläpito sisältää myös palautteiden lukemista, niihin vastaamista ja korjausten tekemistä, kun sivustolle suunniteltu palauteosio on saatu valmiiksi.

6 POHDINTA

Opinnäytetyön tavoitteena oli kehittää web-sivusto Oulun alueen tapahtumille, joka toimisi mahdollisimman automatisoidusti. Automatisointi oli tarkoitus toteuttaa hakemalla tiedot RSS-syötteillä ja päivittämällä ne Cron-tehtävillä tietokantaan, jotta sivuston ylläpitäminen olisi helppoa. Tarkoituksena oli saada kerättyä paljon tapahtumia sivustolle. Sivustosta pyrittiin luomaan responsiivisesti toimiva ja helppokäyttöinen. Käyttäjille haluttiin luoda mahdollisuus kommentoida ja äänestää haluamaansa tapahtumaa, sekä antaa palautetta sivuston palautesivulta. Kaikille tapahtumille pyrittiin asettamaan kategoria, joiden avulla käyttäjät voivat katsella tapahtumia. Lisäksi käyttäjillä olisi mahdollisuus katsella tapahtumia sivustolla olevan kalenterin tai muiden käyttäjien antamien äänien mukaan. Tavoitteina oli julkaista sivusto joulukuussa 2013, jolloin sivustolle asetetut toiminnallisuudet olisi saatu toteutettua sekä tilattu graafinen suunnitelma olisi otettu käyttöön.

Työn toteutukseen käytettiin paljon erilaisia tekniikoita, joista meillä molemmilla oli jo jonkin verran aikaisempaa kokemusta. Sivustoa toteutettaessa joiduimme opiskelemaan melko paljon lisää muun muassa PHP:tä. Aluksi jaoin tehtävät siten, että toinen loi asettelun mobiilisivustolle sekä web-sivustolle ja toinen tutustui RSS-syötteiden tallentamiseen tietokantaan. Kun asettelu ja tietokanta oli saatu luotua jatkettiin kehittämistä tiiviissä yhteistyössä. Projekti eteni näin hyvin ja jotakuinkin aikataulun mukaisesti.

Sivuston responsiivisuus jäi toteuttamatta, koska sen ei katsottu olevan keskeinen osa sivuston toiminnallisuuden kannalta. Sivuston rakenne on tässä vaiheessa sellainen, että se toimii kelpollisesti erilaisilla näyttöpäätteillä ilman responsiivisuuttakin. Kehitystyön aikana jouduimme hieman karsimaan myös muita asettamiamme tavoitteita aikataulun vuoksi. Näin ollen valitsimme toteutettaviksi vain ne ominaisuudet, jotka olivat mielestämme tärkeimmät sivuston toiminnallisuuden kannalta.

Tavoitteita olisi aluksi voinut rajata hieman tarkemmin, sillä ne olivat melko laajat opinnäytetyöksi. Tavoitteista keskeisimmät, kuten sivuston automaattinen tietojen haku ja päivitys, saatiin toteutettua. Myös tapahtumien kategoriat, kalenteri ja äänestäminen saatiin toteutettua niin kuin oli tarkoitus. Osa tavoitteista kuten kommentointi ja palautelomake jäivät toteuttamatta, mutta katsoimme, että ne eivät vielä tässä vaiheessa olleet keskeisiä ominaisuuksia. Myös ulkonäkö on keskeneräinen ja tapahtumia on vain kahdesta lähteestä, mutta tarkoituksena on jatkokehittää sivustoa tulevaisuudessa näiltä osin. Projektin aikana on syntynyt paljon ideoita lisäominaisuuksille, jotka haluamme toteuttaa sivustoon, mutta niitä ei ole ehditty toteuttaa tässä opinnäytetyössä. Osa näistä myöhemmin huomatuista lisäominaisuuksista olisi ollut hyvä listata jo alkuperäisiin tavoitteisiin, kuten esimerkiksi uusien tapahtumien ehdotus sivustolle sekä ylläpitonäkymä.

Opinnäytetyöraportin kirjoittaminen osoittautui haastavammaksi ja aikaa vievämmäksi kuin kuvittelimme. Opinnäytetyöraportin kirjoittamiseen käytetyn ajan vuoksi, sivustolle ei ehditty toteuttaa aivan kaikkia haluttuja ominaisuuksia, mutta pääpiirteittäin tavoitteissa onnistuttiin.

Lopputuloksena saatiin julkaistua hyvin toimiva sivusto, joka on käyttökelpoinen, vaikkakin osittain keskeneräinen. Sivustoa tullaan kehittämään ja lisäämään suunnitellut ominaisuudet. Sivustoa on helppo lähteä jatkokehittämään, koska sille on rakennettu hyvä ja toimiva pohja. Lisäksi on suunniteltu mitä ominaisuuksia sivustolle halutaan luoda seuraavaksi. Opinnäytetyöprojektin aikana tutustuimme joihinkin uusiin menetelmiin, joita voidaan käyttää sivuston jatkokehityksessä, sekä kehityimme web-sivuston kehittäjinä huomattavasti.

LÄHTEET

Haverinen R. 2011. CSS:Kirjoitussäännöt. Hakupäivä 10.10.2013.
<http://raulihaverinen.fi/verkko-opas/?id=83>

HostGator. What are Cron Jobs? Hakupäivä 25.10.2013.
<http://support.hostgator.com/articles/cpanel/what-are-cron-jobs>

Huotari J. 2013. SQL-kielen perusteet. Hakupäivä 11.11.2013.
<http://homes.jamk.fi/~huojo/opetus/IIZO3030/SQLopas.pdf>

Jyväskylän yliopisto. Verkkosivuston käytettävyyden arviointi. Hakupäivä 15.10.2013.
<https://koppa.jyu.fi/avoimet/hum/tvt/sosiaalinen-web/verkkosivuston-kaeytettaevyyden-arviointi>

Järvenpää J. Opi rakentamaan selkeä sivusto. Hakupäivä 15.10.2013.
http://www.sivut.org/sivusto/vinkit/selkeys.php#jatkuvuus_ja_navigaatio

Korpela, J. K. & Linjama T. 2004. XHTML-käsikirja. Jyväskylä: Docendo.

Korpela, J. K. & Linjama T. 2005. Web-suunnittelu. Jyväskylä: Docendo.

Korpela, J. 2009a. Web-julkaisemisen opas. Hakupäivä 15.10.2013.
<http://www.cs.tut.fi/~jkorpela/webjulk/2.1.html>

Korpela, J. 2009b. Web-julkaisemisen opas. Hakupäivä 15.10.2013.
<http://www.cs.tut.fi/~jkorpela/webjulk/2.3.html#runko>

Korpela J.K. 2012. Responsiivinen suunnittelu. Hakupäivä 11.11.2013.
<http://html5kirja.fi/2012/08/02/responsiivinen-suunnittelu/>

Kyrnin. What is CSS? Hakupäivä 10.10.2013.

<http://webdesign.about.com/od/beginningcss/a/aa021607.htm>

Laaksonen A, 2009. MySQL ja PHP: Osa 8 – Tietokannan suunnittelu.

Hakupäivä 25.10.2013.

<http://www.ohjelmointiputka.net/oppaat/opas.php?tunnus=mysqlphp08>

Lister J. 2013. What is an Internet Portal? Hakupäivä 13.11.2013.

<http://www.wisegeek.com/what-is-an-internet-portal.htm>

Narayan S. 2011. What is jQuery and How to Start using jQuery? Hakupäivä

13.11.2013.

<http://www.codeproject.com/Articles/157446/What-is-jQuery-and-How-to-Start-using-jQuery>

Nixon R 2009. PHP, MySQL & JavaScript. Sebastopol CA: O'Reilly.

PHP Group 2013a. What is PHP? Hakupäivä 11.10.2013.

<http://fi1.php.net/manual/en/intro-whatis.php>

PHP Group 2013b. What can PHP do? Hakupäivä 11.10.2013.

<http://fi1.php.net/manual/en/intro-whatcando.php>

Pilgrim 2002. What is RSS. Hakupäivä 1.11.2013.

<http://www.xml.com/pub/a/2002/12/18/dive-into-xml.html>

Saarikumpu O. 2013. Johdanto CSS-tyyliehdotuksien käyttö Web-sivuilla.

Hakupäivä 10.10.2013.<http://weppipakki.com/css/tekstit/cssintro.htm#cyd>

Shapiro 2006. RSS Explained. Hakupäivä 25.10.2013.

Hakupäivä 25.10.2013.<http://rssexplained.blogspot.fi/>

Tatum M. 2011. What is a Web Portal? Hakupäivä 13.11.2013.
<http://www.wisegeek.org/what-is-a-web-portal.htm>

Winkler R. What is a Web Portal? Hakupäivä 13.11.2013.
<http://www.atlanticwebfitters.ca/WhatisaWebPortal/tabid/95/Default.aspx>

W3C 2011. HTML5 differences from HTML4. Hakupäivä 9.10.2013.
<http://www.w3.org/TR/2011/WD-html5-diff-20110405/>