

Saimaan ammattikorkeakoulu  
Tekniikka Lappeenranta  
Tietotekniikka  
Ohjelmistotekniikka

Marko Lähde

## **Nuoren kaupunki -WWW-sivusto**

Opinnäytetyö 2015

## Tiivistelmä

Marko Lähde

Nuoren kaupunki -WWW-sivusto, 41 sivua

Saimaan ammattikorkeakoulu

Tekniikka Lappeenranta

Tietotekniikka

Ohjelmistotekniikka

Opinnäytetyö 2015

Ohjaajat: lehtori Mikko Huhtanen, Saimaan ammattikorkeakoulu, toiminnanjohtaja Jukka Nokelainen, Savonlinnan Seudun Nuorisotoiminnan Tuki ry

Tämän opinnäytetyön tarkoituksena oli kehittää nuoren elämäntilanteita sekä Savonlinnan kaupungin tuki- ja apuverkostoa esittelevä WWW-sivusto. Asiakaina olivat Savonlinnan Seudun Nuorisotoiminnan Tuki ry ja Savonlinnan kaupungin Kanava – työvoimaa avoimille työmarkkinoille -hanke. Sivuston ylläpidosta vastaavan asiakkaan lisäksi loppukäyttäjinä ovat Savonlinnan seudun 16–29-vuotiaat nuoret.

Opinnäytetyössä määriteltiin sivuston vaatimukset, suunniteltiin ja toteutettiin tietokanta ja sivuston käyttöliittymät sekä testattiin ja otettiin käyttöön valmis sivusto. Sivuston toteutuksessa käytettiin muun muassa PHP-, jQuery- ja MySQL-tekniikoita.

Opinnäytetyön tuloksena oli toimiva ja asiakkaan vaatimusten mukainen sivusto.

Asiasanat: PHP, jQuery, MySQL

## **Abstract**

Marko Lähde

Nuoren kaupunki website, 41 pages

Saimaa University of Applied Sciences

Technology Lappeenranta

Information Technology

Software Engineering

Bachelor's Thesis 2015

Instructors: Mr Mikko Huhtanen, lecturer, Saimaa University of Applied Sciences, Mr Jukka Nokelainen, executive director, Savonlinnan Seudun Nuorisotoiminnan Tuki ry

The purpose of this thesis was to develop a website depicting the services the city of Savonlinna has for the young. The website was commissioned by the Savonlinnan Seudun Nuorisotoiminnan Tuki association and the Kanava – työvoimaa avoimille työmarkkinoille project of the city of Savonlinna. Along with the commissioners' the end users of the website are the 16- to 29-year-old people of the Savonlinna area.

This thesis was carried out by specifying the requirements of the website, by designing and implementing the database and the user interfaces and by testing and deploying the final product. PHP, jQuery and MySQL applications and programming languages, among others, were used in the implementation.

The final result of this thesis was a working website that met the commissioners' requirements.

Keywords: PHP, jQuery, MySQL

## Sisältö

Käsitteet.....	5
1 Johdanto.....	7
2 Käytetyt tekniikat.....	7
2.1 HTML ja CSS.....	8
2.2 WWW-palvelin.....	9
2.3 PHP.....	10
2.4 Relaatiotietokanta.....	10
2.5 SQL.....	12
2.6 JavaScript ja jQuery.....	13
2.6.1 ImageMapster.....	13
2.6.2 CKEditor.....	14
2.7 Ajax.....	15
3 Käytetyt työkalut.....	15
3.1 Notepad++.....	16
3.2 PhpMyAdmin.....	16
3.3 GIMP.....	17
3.4 FileZilla.....	17
3.5 Google Street View Image API.....	17
4 WWW-sivuston tietoturva.....	18
4.1 Tietokannan asetustiedosto.....	18
4.2 Salasanoiden tallentaminen.....	18
4.3 SQL-injektio ja valmistellut lausekkeet.....	19
4.4 XSS, syötteiden validointi ja tulosteiden suodatus.....	20
4.5 Asiakas-palvelin-yhteyden suojaaminen.....	20
4.6 Virheilmoitukset.....	21
5 Työn toteutus.....	22
5.1 Asiakas ja taustaa työlle.....	22
5.2 Esitutkimus ja määrittely.....	24
5.3 Sivuston osat.....	26
5.3.1 Tietokanta.....	27
5.3.2 Karttakäyttöliittymä.....	29
5.3.3 Hallintakäyttöliittymä.....	31
5.3.4 Mobiilikäyttöliittymä.....	33
5.4 Testaus.....	34
5.5 Dokumentointi.....	35
5.6 Käyttöönotto.....	36
6 Pohdinta.....	37
Kuvat.....	39
Lähteet.....	40

## Käsitteet

Ajax	Asynchronous JavaScript and XML. Joukko tekniikoita asynkronisen tiedonsiirron mahdollistamiseen WWW-sivuilla.
Avoin lähdekoodi	Ohjelmointitapa, jossa lähdekoodi julkaistaan ilmaiseksi kaikkien nähtäväksi ja muokattavaksi omien tarpeiden mukaan.
CSS	Cascading Style Sheets. Tyyliohje, jolla määritetään WWW-sivun ulkoasu.
DOM	Document Object Model. Oliomalli, joka esittää WWW-sivun rakenteen puumaisessa muodossa.
FTP	File Transfer Protocol. Tiedonsiirtoprotokolla, jolla siirretään tiedostoja palvelimen ja käyttäjän välillä.
HTTP	Hypertext Transfer Protocol. Tiedonsiirtoprotokolla, jolla siirretään WWW-sivuja palvelimen ja käyttäjän välillä.
HTML	Hypertext Markup Language. Kuvaskieli, jolla määritetään WWW-sivun rakenne.
Internet	Paikallisten tietoverkkojen ja -koneiden välinen maailmanlaajuinen tietoverkko.
JavaScript	Selaimessa suoritettava komentosarjakieli, jolla WWW-sivuille voi lisätä dynaamista sisältöä.
jQuery	Avoin JavaScript-kirjasto.
MySQL	Relaatiotietokannan hallintajärjestelmä.
Nutu	Savonlinnan Seudun Nuorisotoiminnan Tuki ry.
PHP	PHP: Hypertext Preprocessor. WWW-sivuilla käytettävä palvelinpuolen komentosarjakieli.
Responsiivisuus	Suunnittelumenetelmä, jonka avulla sivusto mukautuu automaattisesti erikokoisille päätelaitteille.
SQL	Structured Query Language. Tietokantojen kyselykieli.
URL	Uniform Resource Locator. Osoittaa tiedoston, esimerkiksi WWW-sivun sijainnin internetissä.

W3C	World Wide Web Consortium. WWW-standardeja kehittävä ja ylläpitävä organisaatio.
Webhotelli	Kiintolevytilaa ja muita verkkopalveluita WWW-sivuja varten tarjoava palvelu.
Verkkotunnus	Palvelimen tai palvelinryhmän internetissä yksilöivä selkokielen tunnus.
WWW	World Wide Web. Internetissä toimiva palvelu, jossa palvelimet jakavat sisältöä käyttäjille.
WYSIWYG	What You See Is What You Get. WYSIWYG-editori on tekstieditori, jossa tekstin ulkoasu on samanlainen sekä muokatessa että valmiissa lopputuloksessa.

## **1 Johdanto**

Tämän opinnäytetyön tarkoituksena on kehittää nuoren elämäntilanteita sekä Savonlinnan kaupungin tuki- ja apuverkostoa esittelevä WWW-sivusto. Asiakkaina ovat Savonlinnan Seudun Nuorisotoiminnan Tuki ry, joka vastaa työn päätyttyä sivuston ylläpidosta ja tietojen päivittämisestä, ja Savonlinnan kaupungin Kanava – työvoimaa avoimille työmarkkinoille -hanke. Ylläpitäjän ohella varsinaisina loppukäyttäjinä ovat Savonlinnan seudun 16–29-vuotiaat nuoret, jotka voivat hyödyntää sivustoa etsiessään apua eri elämäntilanteissa kohtaamiinsa ongelmiin. Sivusto edistää näin omalta osaltaan nuorisotakuun toteutumista.

Työn lähtökohtana on, että aiempaa sivustoa ei ole olemassa, joten tietokanta ja käyttöliittymät on suunniteltava alusta alkaen. Tavoitteena on helppokäyttöinen ja asiakkaan helposti päivitettävissä oleva sivusto, joka kokoaa tiedot nuoren kannalta hyödyllisistä palveluista yhteen paikkaan. Sivuston tulee toimia myös mobiililaitteissa.

Työ rajataan siten, että asiakkaalle toimitetaan valmis, käyttöön otettu sivusto ohjeineen. Opinnäytetyön tekijä vastaa sivuston käyttöönotosta asiakkaan määrämällä palvelimella.

Tämän raportin teoriaosuudessa esitellään lyhyesti työssä käytettyjä menetelmiä, tekniikoita ja työkaluja sekä pohditaan muutamia WWW-sivuston tietoturvakysymyksiä. Toteutusosuudessa perustellaan käytetyt ratkaisut ja kuvataan työn toteutus vaihe vaiheelta.

## **2 Käytetyt tekniikat**

Tässä luvussa esitellään työn toteutuksessa käytettyjä tekniikoita. Käytetyt tekniikat ovat yleisiä WWW-tekniikoita ja vapaasti saatavilla olevia avoimeen lähdekoodiin perustuvia ratkaisuja. Valintoihin vaikuttivat sekä projektin budjetti että sivuston sijoituspaikaksi valitun webhotellin ominaisuudet.

## 2.1 HTML ja CSS

Hypertext Markup Language eli HTML on kuvauskieli, jolla muodostetaan WWW-sivun rakenne. HTML-dokumentti koostuu tekstistä, jonka sekaan lisätään sivun rakenteeseen vaikuttavia komentoja eli tageja. Tageilla voidaan esimerkiksi määrittää tietty osa tekstistä otsikoksi ja tietty leipätekstiksi, luoda taulukoita ja linkittää toisiin WWW-sivuihin. Tagien avulla sivulle voi myös liittää kuvia ja muita multimediaelementtejä. Tagi koostuu aloitus- ja lopetustagista, ja yhdessä ympäröimänsä sisällön kanssa ne muodostavat elementin. Jokainen elementti on aina toisen elementin sisällä, lukuun ottamatta ylintä html-elementtiä. (Heinisuo & Rauta 2007, 252–253; Paananen 2001, 236–237)

HTML pohjautuu Standard Generalized Markup Languageen eli SGML:ään, jolla määritellään muita kuvauskieliä. HTML:n sukulauskieli on Extensible Markup Language eli XML, joka on yleiskäyttöinen kuvauskieli. XML:ssä ei ole HTML:n tapaan ennalta määrättyjä tageja, vaan sillä voidaan kuvata mikä tahansa kuvailtavissa oleva asia. Tunnettuja sovellutuksia ovat SVG-vektorigrafiikka ja RSS-syötteen. XML on HTML:ää huomattavasti tiukemmin määritelty merkintöjen oikeaoppisuuden suhteen. Extensible Hypertext Markup Language eli XHTML on HTML:n seuraaja, joka noudattaa XML:n muotoilusääntöjä. (Heinisuo & Rauta 2007, 250–251)

Document Object Model eli DOM on oliomalli, jolla kuvataan HTML-dokumentin rakenne puumaisessa muodossa. Jokainen dokumentin elementti muodostaa oman olion, jolla on jäsenmuuttujia ja jäsenfunktioita. DOM mahdollistaa HTML-dokumentin tutkimisen ja käsittelemisen esimerkiksi JavaScriptin avulla. (Heinisuo & Rauta 2007, 253–254)

Cascading Style Sheets eli CSS on HTML:n rinnalla käytettävä tyyliohje, joka määrittää rakenteisen HTML-sivun eri elementtien ulkoasun, kuten värin, sijoittelun ja fontin. Se ei kuitenkaan ole sidonnainen pelkästään HTML:ään, vaan sitä voidaan käyttää minkä tahansa XML-pohjaisen kielen kanssa. WWW-sivun rakenteen ja ulkoasun erottaminen toisistaan helpottaa sivujen ylläpitoa ja sivujen räätälöintiä erilaisiin ympäristöihin, kuten työpöytäkäyttöön, mobiililaitteisiin ja tulosteisiin. (W3C 2015)



HTML, CSS ja DOM ovat standardoituja tekniikoita, joita ylläpitää ja kehittää World Wide Web Consortiumin eli W3C (W3C 2015).

Esimerkki HTML-koodista:

```
<html>
  <head>
    <title>Loru</title>
  </head>
  <body>
    <h1>Loru</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. In
    pulvinar.</p>
  </body>
</html>
```

Esimerkki CSS-koodista:

```
h1 {
  font-size: 4em;
}

p {
  margin-left: 1em;
}
```

## 2.2 WWW-palvelin

WWW-palvelin on verkkoon liitetty tietokone sekä tietokoneella oleva ohjelma, joka kommunikoi toisten tietokoneiden, asiakkaiden, kanssa HTTP-protokollan välityksellä. Palvelin välittää asiakkaan pyyntöjen perusteella WWW-sivuja ja muita tiedostoja, jotka luetaan asiakastietokoneella WWW-selaimella. (Heinisuo & Rauta 2007, 12)

WWW-palvelinohjelman voi asentaa omalle tietokoneelle, mutta usein käytetään ulkopuolista palvelua, kuten niin sanottua webhotellia, jossa on myös muut WWW-sivujen ylläpitoon tarvittavat ominaisuudet. Ulkopuolisen palvelun etuna on, että tällöin palvelimen itsensä tietoturva, asetukset ja päivitykset ovat palveluntarjoajan vastuulla. Toisaalta palveluntarjoajan tekemät asetukset ja päivitykset voivat aiheuttaa odottamattomia rajoituksia ja ongelmia www-sivujen toiminnalle. (Heinisuo & Rauta 2007, 14)

Yleisesti käytettyjä WWW-palvelinohjelmia ovat muun muassa Apache ja Microsoftin Internet Information Services eli IIS (Heinisuo & Rauta 2007, 15).

### 2.3 PHP

PHP: Hypertext Preprocessor eli PHP on ohjelmointikieli, jota käytetään yhdessä WWW-palvelimen ja tietokannan kanssa WWW-sovellusten luomiseen (Heinisuo & Rauta 2007, 11–14). PHP-koodia kirjoitetaan tyypillisesti suoraan HTML-koodin sekaan. PHP on tyypiltään tulkittava palvelinpuolen komentosarjakieli eli PHP-koodi suoritetaan ja tulkitaan palvelimella joka kerta ennen kuin palvelin lähettää valmiin sivun asiakasseläimeen. PHP-tulkki on yleensä liitetty erillisenä moduulina WWW-palvelimeen. (Heinisuo & Rauta 2007, 11–14) Erään tilaston mukaan PHP on käytössä yli 80 prosentissa jotain palvelinpuolen ohjelmointikieltä käyttävistä sivustoista (W3Techs 2015).

PHP:ssä on mukana tavanomaiset ohjelmointikielten ominaisuudet, kuten muuttujat, ohjausrakenteet, taulukot ja funktiot. Perinteisen proseduraalisen ohjelmoinnin lisäksi PHP:tä voi käyttää oliopohjaiseen ohjelmointiin. PHP:n toiminnallisuutta on mahdollista laajentaa ohjelmakirjastoilla, joita siihen sisältyy lukuisia jo valmiiksi. Muun muassa tietokantojen käsittely PHP:n valmiiden ohjelmakirjastojen avulla on yksinkertaista. (Heinisuo & Rauta 2007, 11–14) Syntaksiltaan PHP muistuttaa esimerkiksi C-, Perl- ja Java-kieliä (Heinisuo & Rauta 2007, 64).

Esimerkki PHP-koodista:

```
<?php
    $nimi = "Marko";
    $pvm = date("l");
    echo "Hei, $nimi! Tänään on $pvm.";
?>
```

### 2.4 Relaatietietokanta

Relaatietietokanta on tietovarasto, jossa tieto on tallennettu riveistä eli tietueista ja sarakkeista eli kentistä koostuviin tauluihin. Taulut muodostavat yhteyksiä toisiinsa tietosisältöjensä mukaisilla viittauksilla. (Paananen 2001, 311–312) Kuvassa 2.1 on esimerkki taulurakenteesta.

asiakas			varaus			
asiakasno	nimi	osoite	varausno	asiakasno	päiväys	klo
101	Elli	Oulu	1	101	2000-02-22	12.00
102	Olli	Vihti	2	103	2000-03-13	10.00
103	Alli	Pori	3	101	2000-05-04	15.00
			4	101	2000-05-05	14.00

Kuva 2.1. Esimerkki tietokannan tauluista (Paananen 2001, 311).

Sarakkeella on aina nimi ja tietotyyppi, joka määrittää millaista tietoa siihen voi tallentaa. Yleisimmät tietotyypit voidaan jakaa kolmeen luokkaan: numero-, teksti- ja aikatyypit. (Paananen 2001, 318)

Nimen ja tietotyypin lisäksi sarakkeella voi olla lisämääreitä, joista tärkeimmät ovat perus- ja vierasavain. Kullekin tietokannan taululle määritetään perusavain (primary key), joka yksilöi taulun rivit. Esimerkin asiakas-taulussa perusavain on asiakasno-sarake ja varaus-taulussa varausno-sarake. Myös Varaus-taulussa on asiakasno-sarake, joka on asiakas-taulun asiakasno-sarakkeeseen viittaava viiteavain (foreign key). Perus- ja viiteavaimet muodostavat eheyssäännön: kukin perusavain saa esiintyä taulussa vain kerran, ja viiteavaimen on aina viitattava jonkin toisen taulun avaimen. Jos asiakas-taulusta yritetään poistaa rivi, johon viitataan varaus-taulusta, seuraa virheilmoitus. Toisin sanoen asiakkaan poistamiseksi täytyy poistaa ensin asiakkaaseen liittyvät varaukset. Vastaavasti lisäykset on tehtävä siten, että asiakas on olemassa ennen asiakkaaseen viittaavaa varausta. (Paananen 2001, 311–312)

Tiedon jakamista erillisiin tauluihin itsenäisten käsitteiden (asiakas ja varaus) perusteella siten, että kunkin sarakkeen arvo esittää pienimmistä perusosista koostuvaa tietoa, kutsutaan normalisoinniksi. Normalisointi vähentää turhaa toistoa, kun samaa tietoa ei tarvitse lisätä ja päivittää useaan tauluun ja edesauttaa näin myös tiedon eheyttä. (Paananen 2001, 319–320)

Tietokannan ylläpitoon käytettävää ohjelmistoa kutsutaan tietokannanhallintajärjestelmäksi. Tunnettuja järjestelmiä ovat muun muassa MySQL, PostgreSQL, Sybase, Oracle ja Microsoft SQL Server (Heinisuo & Rauta 2007, 39 ja 274). Tässä työssä käytetyn MySQL:n etuja ovat muun

muassa suorituskyky, tuki useille ohjelmointikielille, asennuksen ja ylläpidon helppous, kattava dokumentaatio sekä edullisuus myös yrityskäytössä (Heinisuo & Rauta 2007, 37–39).

Relaatiomallin isänä pidetty E. F. Codd esitti vuonna 1982 tietokannanhallintajärjestelmälle kahdeksan perustehtävää, joita noudatetaan edelleen (Paananen 2001, 308–309):

1. Tallentaminen, haku ja päivitykset
2. Systeemitaulusto
3. Eheysvalvonta
4. Transaktion hallinta
5. Samanaikaisuuden hallintaan
6. Toipuminen vikaantumisesta
7. Tietojen suojauspalvelu
8. Tietoliikennepalvelut.

## **2.5 SQL**

Structured Query Language eli SQL on ISO-standardoitu relaatiotietokantojen määrittelyyn ja käsittelyyn käytettävä kieli. SQL-kieli koostuu komentosanoista, jotka jakautuvat määrittely- ja käsittelykomentoihin. Määrittelykomennolla luodaan uusia tietorakenteita ja käsittelykomennolla käsitellään tietokantaan tallennettua tietoa. Esimerkki määrittelykomennosta on CREATE TABLE, jolla luodaan uusi taulu tietokantaan. Tärkeimmät määrittelykomennot ovat SELECT, UPDATE, INSERT ja DELETE, joilla voidaan hakea, päivittää, lisätä ja poistaa, tässä järjestyksessä, taulun tietoja. (Heinisuo & Rauta 2007, 97; Paananen 2001, 316–317)

Esimerkkejä SQL-lausekkeista (Paananen 2001, 316–317):

- CREATE TABLE asiakas (asiakasno int, nimi char(30), osoite (char(40), PRIMARY KEY (asiakasno))
- SELECT asiakasno, päiväys, klo FROM varaus WHERE asiakasno = 101 OR klo <= '12.00'
- UPDATE varaus SET klo = '9.00' WHERE asiakasno = 101
- DELETE FROM varaus WHERE asiakas = 101
- INSERT INTO varaus VALUES (5, 102, '2001-03-11', '11.00')

## 2.6 JavaScript ja jQuery

JavaScript on ohjelmointikieli, jolla WWW-sivuille voi lisätä käyttäjän toimiin reagoivaa vuorovaikutteisuuutta. JavaScript on PHP:n tapaan tulkittava kieli, mutta PHP:stä poiketen koodi suoritetaan ja tulkitaan suoraan käyttäjän WWW-selaimessa. Tulkkaamisesta vastaa selaimen oma JavaScript-tulkki. JavaScript-koodi toimitetaan selaimelle joko HTML-koodin lomassa script-elementin sisällä tai erillisessä tiedostossa, johon viitataan script-elementillä. (Heinisuo & Rauta 2007, 254)

jQuery on suosittu ja monipuolinen JavaScript-kirjasto, joka helpottaa JavaScriptin käyttöä valmiiden metodien avulla. jQuery mahdollistaa muun muassa HTML-elementtien manipuloinnin, tapahtumien hallinnan, animaatioiden luomisen ja Ajax-kutsut. jQueryyn liittyvät sisarprojektit jQuery UI, joka on kokoelma valmiita käyttöliittymäinteraktioita, -efektejä ja -vimpaimia ja jQuery Mobile, joka mahdollistaa mobiililaitteille optimoitujen responsiivisten WWW-sivujen toteuttamisen. (jQuery API 2015)

### 2.6.1 ImageMapster

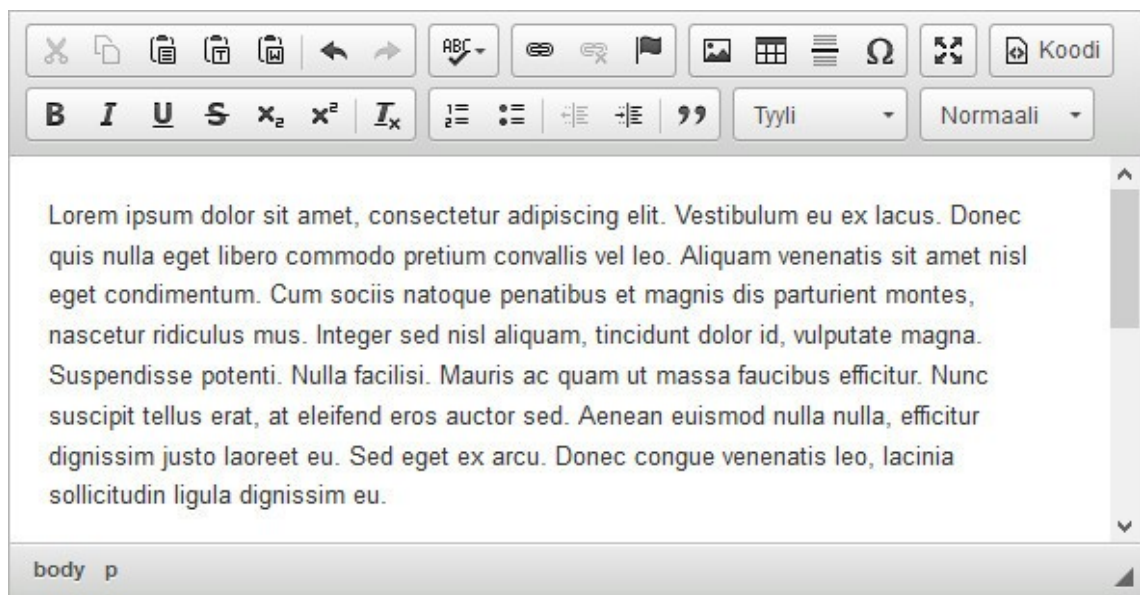
ImageMapster on jQuery-liitännäinen, jolla HTML:n kuvakartan voi muuttaa interaktiivisemmaksi (ImageMapster 2015). HTML:n map- ja area-elementeillä toteutettava kuvakartta (image map) on kuva, joka sisältää ympyrän, suorakulmion tai monikulmion muotoisia napsautettavia alueita, jotka toimivat linkkeinä muille sivuille. ImageMapsterin avulla kuvakartan alueet voi muuttaa

valittavaksi tai automaattisesti korostuviksi, alueiden kohdalla voi näyttää mukautetun vihjetekstin (tooltip) ja kuvakartan voi liittää ulkoiseen luetteloon (ImageMapster 2015).

Kuvakartan napsautettavat alueet määritetään koordinaatteina alkuperäisen kuvan koon mukaan. Mikäli kuva halutaan näyttää toisessa koossa, osaa ImageMapster automaattisesti skaalata kuvakartan täsmäämään kuvan uuteen kokoon (ImageMapster 2015).

## 2.6.2 CKEditor

CKEditor on pääosin JavaScriptillä toteutettu avoimeen lähdekoodiin perustuva WYSIWYG-editori. Sen avulla voi tuoda tavallisen tekstinkäsittelyohjelman toimintoja suoraan WWW-sivulle helpottamaan sisällöntuotantoa. Tekstin muotoilun lisäksi CKEditorilla voi muun muassa lisätä sivulle linkkejä, kuvia, animaatioita ja lomake-elementtejä. Se tukee tekstin liittämistä suoraan Wordista ja oikolukua. Varsinaiset muotoilut tehdään XHTML-koodilla, jonka voi tallentaa sellaisenaan tietokantaan. (CKEditor 2015) Kuvassa 2.2 on CKEditor perusasetuksilla.



Kuva 2.2 CKEditor.

CKEditor on mukautettavissa asetustiedostoa muokkaamalla ja laajennettavissa liitännäisillä, joita on kaikkiaan yli 300. Editorin itsensä saa liitännäisenä muun

muassa suosittuihin sisällönhallintajärjestelmiin Drupaliin, Joomlaan ja Wordpressiin. (CKEditor 2015)

## **2.7 Ajax**

Ajax on joukko tekniikoita, joiden avulla WWW-selain voi vaihtaa tietoa palvelimen kanssa ja päivittää osia WWW-sivusta ilman, että koko sivua tarvitsee ladata uudelleen. Ajax on alun perin lyhenne sanoista Asynchronous JavaScript and XML. Esimerkkejä Ajaxia hyödyntävistä palveluista ovat Googlen Gmail ja Google Maps. (Ullman 2007)

Ajaxiin liittyviä tekniikoita (Ullman 2007):

- (X)HTML ja CSS, sivun rakenteen ja ulkoasun määrittelyyn
- DOM, mahdollistamaan HTML-dokumentin dynaaminen käsittely
- XML, palvelimelta välitettävän tiedon muotoiluun. Käytännössä tieto voi olla myös HTML- tai JSON-muodossa tai pelkkää tekstiä.
- XMLHttpRequest, selaimen sisäänrakennettu JavaScript-olio asynkronisen kommunikoinnin toteuttamiseen.
- JavaScript, DOM:n dynaamisen käsittelyyn ja XMLHttpRequest-olion luontiin.

Käytännössä myös palvelinpuolen ohjelmointikielen, kuten PHP:n, käyttö on tarpeellista esimerkiksi tietokannan käsittelyä varten (Ullman 2007).

## **3 Käytetyt työkalut**

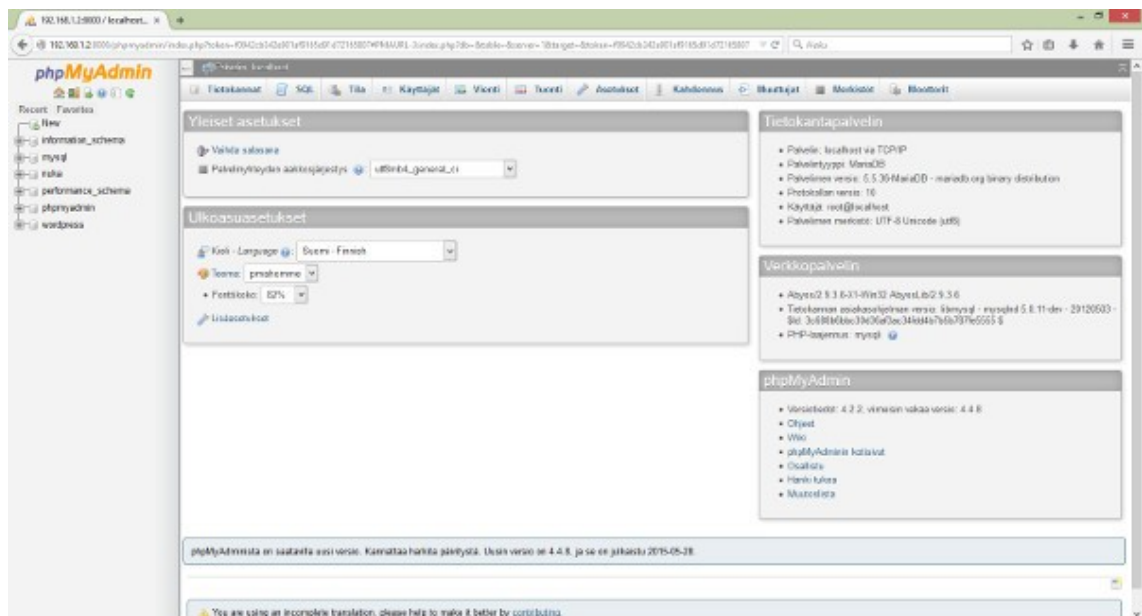
Tässä luvussa esitellään työn toteutuksessa käytettyjä työkaluja. Edellä mainittujen tekniikoiden tapaan myös käytetyt työkalut ovat pääosin avoimeen lähdekoodiin perustuvia ratkaisuja.

### 3.1 Notepad++

Notepad++ on avoimeen lähdekoodiin perustuva kevyt teksti- ja lähdekoodieditori Windows-käyttöjärjestelmälle. Sen ominaisuuksiin kuuluvat muun muassa välilehtinäköymä, joka mahdollistaa useamman tekstitiedoston avaamisen samassa ikkunassa, ja lähdekoodin värikorostus, joka tekee koodista helppolukuisempaa.

### 3.2 PhpMyAdmin

PhpMyAdmin on avoimeen lähdekoodiin perustuva PHP:llä toteutettu MySQL-tietokannanhallintatyökalu, jota käytetään WWW-selaimen kautta. PhpMyAdminissa on tuki useimmille MySQL-tietokannan perustoiminnoille, kuten tietokannan, taulun, kentän, näkymän ja indeksin luominen, muokkaaminen ja poistaminen, SQL-lausekkeiden suorittaminen sekä käyttäjien ja käyttöoikeuksien hallinta. Sillä voi myös muun muassa viedä ja tuoda tietoja eri formaateissa ja luoda graafisia esityksiä tietokannan rakenteesta. (PhpMyAdmin 2015) Kuvassa 3.1 on PhpMyAdminin käyttöliittymä.



Kuva 3.1. PhpMyAdminin käyttöliittymä.



### 3.3 GIMP

GNU Image Manipulation Program eli GIMP on avoimeen lähdekoodiin perustuva kuvankäsittelyohjelma bittikarttagrafiikalle. GIMP tarjoaa ilmaisen vaihtoehdon muun muassa Adobe Photoshopille.

### 3.4 FileZilla

FileZilla on avoimeen lähdekoodiin perustuva alustariippumaton FTP-ohjelma. Se tukee tavallisen FTP-tiedonsiirron lisäksi suojattua SFTP- ja FTPS-tiedonsiirtoa. Asiakasohjelma FileZillan lisäksi on saatavilla palvelinohjelma FileZilla Server.

### 3.5 Google Street View Image API

Google Street View Image API on Googlen tarjoama HTTP-rajapinta, jolla voi muodostaa staattisen kuvan Googlen karttapalvelun katunäkymästä yksinkertaisen URL-osoitteen avulla. (Google Developers 2015)

Osoite on perusmuodoltaan seuraavanlainen:

<https://maps.googleapis.com/maps/api/streetview?parametrit>

Haluttu kuva määritetään osoitteeseen liitettävien parametrien avulla. Esimerkiksi katunäkymäkuva Saimaan ammattikorkeakoulun Skinnarilan kampuksesta muodostetaan seuraavalla osoitteella:

<https://maps.googleapis.com/maps/api/streetview?size=400x400&location=61.066432,28.093568&heading=240>

jossa *size* määrittää kuvan koon pikseleinä, *location* kameran sijainnin koordinaatteina ja *heading* kameran suunnan asteina. Sijainti on mahdollista antaa myös katuosoitteen avulla. (Google Developers 2015)

Palvelu on peruskäytössä ilmainen, mutta Google suosittelee sen käyttäjille Google-tiliin liitettävän API-avaimen luomista. (Google Developers 2015)

## 4 WWW-sivuston tietoturva

Julkisen luonteensa vuoksi WWW-sivustot ovat alttiita monenlaisilla tietoturva-aukoille ja hyökkäyksille. Tässä luvussa käydään läpi muutamia tietoturvauhkia ja keinoja niihin varautumiseen lähinnä PHP-kielen ja MySQL-tietokannan näkökulmasta.

### 4.1 Tietokannan asetustiedosto

MySQL-tietokannan käsittely PHP:n välityksellä vaatii tietokannan osoitteen, nimen, käyttäjätunnuksen ja salasanan osana tietokantayhteyden avausfunktia. Mikäli käytössä on oma WWW-palvelin, nämä tiedot ovat useimmiten mahdollista määritellä turvallisesti palvelimen omassa asetustiedostossa. Jos kuitenkin toimitaan jaetulla palvelimella, kuten webhotellissa, tiedot tallennetaan tyypillisesti osaksi tietokantaa käyttävän sivuston lähdekoodia. Tämä altistaa tietojen paljastumiselle ulkopuolisille, jos palvelin esittää PHP-tiedostoja sellaisenaan tulkkamattomina esimerkiksi virheellisen konfiguroinnin vuoksi.

Hyvä ratkaisu on laittaa tiedot erilliseen tiedostoon, joka sijoitetaan WWW-juurihakemiston ulkopuolelle, mistä se sisällytetään tarvittaessa PHP:n include-komennolla. Apache-palvelimessa ja useissa muissa palvelinohjelmistoissa voi käyttää myös hakemistokohtaista .htaccess-tiedostoa, jolla tiedosto suojataan käyttöltä:

```
<files asetustiedosto.php>  
    order allow,deny  
    deny from all  
</files>
```

### 4.2 Salasanojen tallentaminen

Sivuston käyttäjien salasanoja ei tulisi koskaan tallentaa tietokantaan selkokielistä, vaan jonkin salausalgoritmin avulla muodostettuina tiivisteinä. Tällöin tietojen joutuessa vääriin käsiin tietomurron yhteydessä ei kaikki vielä ole menetettyä, sillä tiivisteiden purkamiseksi vaaditaan sanakirja- tai brute force-hyökkäystä eli käytännössä jokaisen mahdollisen vaihtoehdon läpikäymistä, kunnes oikea salasana löytyy. Mikäli salasanaan lisätään vielä satunnainen

merkkijono eli niin sanottu suola, muuttuu tehtävä jo hyvin paljon aikaa ja laskentatehoa vaativaksi. Myös salasanojen itsessään tulisi olla riittävän monimutkaisia eli sisältää yhteensä vähintään kahdeksan merkkiä, pieniä ja isoja kirjaimia sekä numeroita. (Gerber & Compton 2014)

PHP:n versiossa 5.5 ja uudemmissa on helppokäyttöinen `password_hash()`- ja `password_verify()`-funktiot sisältävä salasanatiivisterajapinta. `Password_hash()` luo sille parametrina annetusta selkokieelisestä salasanasta automaattisesti suolatun tiivisteen käyttäen vahvaa `bcrypt`-algoritmia. `Password_verify()`:lla puolestaan voi suoraan vertailla esimerkiksi tietokannasta haettua tiivistettä annettuun selkokieeliseen salasanaan.

### **4.3 SQL-injektio ja valmistellut lausekkeet**

Tietokantaan tehtävät SQL-kyselyt rakennetaan usein osittain käyttäjän syötteiden perusteella, mutta käyttäjän syötteisiin ei koskaan saa luottaa sellaisenaan. SQL-injektio on hyökkäys, jossa lomakekentän tai URL-parametrin välityksellä SQL-kyselyn perään ujutetaan toinen, pahantahtoinen SQL-kysely. Näin järjestelmän toimintaa yritetään manipuloida niin, että tietoa paljastuu tai tuhoutuu. (Gerber & Compton 2014)

PHP:ssä SQL-injektion voi välttää esimerkiksi korvaamalla (`escape`) merkkijonon erikoismerkit `mysql_real_escape_string`-funktiolla ja tarkastamalla numeroarvot `intval`-funktiolla ennen niiden sijoittamista SQL-lausekkeeseen, mutta turvallisempi ratkaisu on käyttää niin sanottuja valmisteltuja lausekkeita. Valmistelluissa lausekkeissa luodaan aluksi lausekemalline, johon käyttäjältä saatavien arvojen paikalle sijoitetaan aluksi pelkät apumerkinnyt. PHP:n MySQLi-rajapinnan kanssa apumerkinnytänä käytetään kysymysmerkkiä. Lausekemalline lähetetään valmisteltavaksi tietokannalle, joka tarkastaa sen syntaksin ja tallentaa sen myöhempää käyttöä varten. Toisessa vaiheessa halutut arvot sidotaan apumerkinntöjen paikalle halutussa tietotyypissä ja lauseke suoritetaan. Valmisteltuja lausekkeita käytettäessä jokaista tietokannan käsittelyyn käytettävää arvoa ei siis tarvitse erikseen siistiä, vaan tietokanta tekee sen automaattisesti. Toisena etuna on tehokkuus peräkkäisissä

kyselyissä, joissa vain arvot muuttuvat. (Heinisuo & Rauta 2007, 274 ja 400–402; PHP Manual 2015b)

Esimerkki valmistellusta lausekkeesta PHP:n MySQLi-rajapinnan kanssa:

```
$lauseke = $yhteys->prepare("UPDATE palvelut SET nimi = ?, kuvaus = ?  
WHERE palveluid = ?");  
$lauseke->bind_param("ssi", $nimi, $kuvaus, $palveluid);  
$lauseke->execute();  
$lauseke->close();
```

#### **4.4 XSS, syötteiden validointi ja tulosteiden suodatus**

SQL-injektiolta suojautumisen ohella käyttäjän syöttämät tiedot täytyy validoida myös tiedon eheyden kannalta. Esimerkiksi pakollisiksi määritellyt kentät on täytettävä, numerokenttiin on oltava mahdollista syöttää vain numeroita ja sähköpostiosoite on annettava oikeassa muodossa. Nämä tarkastukset on mahdollista tehdä selaimessa JavaScriptin avulla, mutta myös palvelinpäässä on reagoitava virheellisiin syötteisiin esimerkiksi siltä varalta, että käyttäjän selaimesta on asetettu JavaScript pois päältä. (Gerber & Compton 2014)

Cross-site scripting eli XSS on tietoturva-aukko, jossa hyökkääjä yrittää ujuttaa lomake-elementin tai URL-parametrin välityksellä haitallista koodia sivustolle (Gerber & Compton 2014). XSS:ltä voi suojautua esimerkiksi muuntamalla HTML:lle varatut erikoismerkit PHP:n htmlspecialchars()-funktiolla niitä vastaaviksi merkkiviittauksiksi (esimerkiksi pienempi kuin -merkistä < tulee &lt;), jolloin niitä ei yritetä tulkita HTML-koodina, vaan tulostetaan sellaisenaan. Tarvittavat muunnokset on parasta tehdä vasta tulostusvaiheessa, sillä jo ennen tietokantaan syöttämistä tehty muunnos voi korruptoida täysin validiakin tietoa ja täten hankaloittaa tiedonhakuja ja tulostamista erilaisiin ympäristöihin. Jos HTML:ää on käytetty tekstin muotoiluun, kuten WYSIWYG-editorit tekevät, ja sen tulkitseminen on siksi toivottua, voi käyttää esimerkiksi valmista HTML Purifier -kirjastoa vaarallisten tagien suodattamiseen.

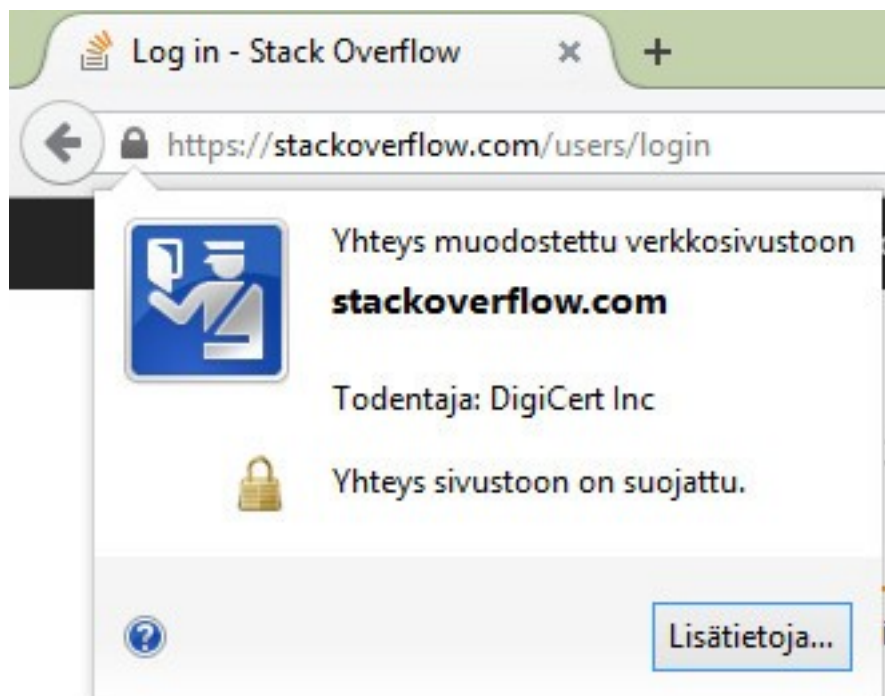
#### **4.5 Asiakas–palvelin-yhteyden suojaaminen**

Asiakkaan ja palvelimen välinen tietoliikenne on yleensä suojaamatonta, mikä voi altistaa muun muassa käyttäjän salasanalan paljastumiselle

sisäänkirjautumisen yhteydessä, jos joku ulkopuolinen taho salakuuntelee yhteyttä (Gerber & Compton 2014). Erityisen helppoa kuuntelu on avoimessa WLAN-verkossa.

Sivuston ylläpitäjä voi hankkia TLS-varmenteen, jonka avulla palvelin autentikoidaan ja yhteys salataan. Tavanomainen HTTP-protokolla toimii tällöin Transport Layer Securityn eli TLS-protokollan päällä, mistä käytetään myös nimitystä HTTPS. TLS:n edeltäjä on Secure Sockets Layer eli SSL.

Suojatun yhteyden tunnistaa WWW-selaimen osoiterivillä näkyvästä lukosta ja osoitteen https-etuliitteestä http:n sijaan, kuten kuvassa 4.1.



Kuva 4.1. Suojatun yhteyden kuvake ja tiedot.

#### 4.6 Virheilmoitukset

Sivuston kehitysvaiheessa on hyödyllistä näyttää selaimessa virheilmoituksia PHP:n havaitsemista ohjelmointivirheistä. PHP:ssä on tähän liittyen kaksi asetusta, `display_errors` ja `error_reporting`, joiden arvoa voi muuttaa joko PHP:n omassa asetustiedostossa tai itse sivuston koodissa. Ensimmäisellä määritellään näytetäänkö virheilmoituksia ja jälkimmäisellä tarkemmin mitä virheilmoituksia näytetään. Siirrettäessä sivusto tuotantokäyttöön

virheilmoitukset on kuitenkin ehdottomasti asetettava pois päältä, sillä ne voivat mahdollisesti paljastaa sivuston heikkouksia. (Heinisuo & Rauta 2007, 36–37)

Oma lukunsa on käyttäjälle virheellisistä syötteistä esitettävät virheilmoitukset. Liian täsmälliset ilmoitukset voivat helpottaa hyökkääjän tehtävää. Esimerkiksi epäonnistuneen sisäänkirjautumisen yhteydessä tulisi näyttää vain yleisluontoinen ilmoitus, jossa ei kerrota täsmälleen mikä tiedoista oli väärin. (Gerber & Compton 2014)

## **5 Työn toteutus**

Tässä luvussa kuvataan työn toteutus vaihe vaiheelta. Sivuston suunnitteluvaiheessa käytettiin paljon protoilua toimivien ratkaisujen löytämiseksi. Lopullista sivustoa rakennettiin inkrementoiden, kunnes sivusto oli valmis.

Prototyypimalli ja inkrementaalinen malli ovat ohjelmiston elinkaaren vaihejakomalleja. Ne ovat joustavampia vaihtoehtoja perinteiselle vesiputousmallille, jonka vaiheet ovat jotakuinkin seuraavat: esitutkimus → määrittely → suunnittelu → toteutus → integrointi ja testaus → käyttöönotto ja ylläpito. (Haikala & Märijärvi 2002, 36) Prototyypimallissa ohjelmiston piirteitä kokeillaan ennen lopullisen ohjelmiston kehittämistä. Prototyypin pohjalta joko määritellään lopullinen ohjelmisto, joka kehitetään alusta alkaen, tai prototyypistä itsestään kehitetään valmis ohjelmisto. (Haikala & Märijärvi 2002, 44) Inkrementaalisisessa mallissa lopullista ohjelmistoa kehitetään projektin sisällä pienissä osissa, inkrementeissa. Jokaisen inkrementin tuloksena on toimiva järjestelmä. (Haikala & Märijärvi 2002, 44)

### **5.1 Asiakas ja taustaa työlle**

Opinnäytetyön asiakkaina olivat Savonlinnan Seudun Nuorisotoiminnan Tuki ry eli Nutu ja Savonlinnan kaupungin Kanava – työvoimaa avoimille työmarkkinoille -hanke. Nutu on Savonlinnan seutukunnassa toimiva yhdistys, joka tuottaa nuorille tietoja, palveluja, toimintoja ja aktiviteetteja (Nutu 2015). Kanava-hanke oli 1.1.2011–31.12.2014 toiminut Euroopan sosiaalirahaston rahoittama Savon-

linnan kaupungin hanke, joka tuki nuorten työllistymistä ja koulutukseen ohjaamista tehostetun palveluohjauksen avulla (Savonlinna 2015a).

Asiakkaan edustajana opinnäytetyöprojektissa toimi Nutun toiminnanjohtaja Jukka Nokelainen. Nutu vastaa opinnäytetyössä tuotetun sivuston ylläpidosta ja päivityksestä työn päättymisen jälkeen. Kanavan puolelta asioita johti projektipäällikkö Virpi Leskinen.

Alkuperäinen idea projektiin tuli Savonlinnan kaupungin nuorten asian yhteisöryhmältä eli Nepparilta, joka antoi sen Kanava-hankkeen kautta nuorten yrittäjyystalo Innostimen ryhmälle käytännön ideointia ja toteutusta varten. Neppari on Nutun koordinoima monialainen yhteistyöryhmä, johon kuuluu 14 eri asiantuntijatahoa Savonlinnan seudulta (Savonlinna 2015b). Innostin on nuorten yrittäjyystalo, joka tarjoaa yrittäjyysvalmennusta ja projekteja Savonlinnan alueen toisen asteen opiskelijoille ja alle 30-vuotiaille työttömille (Innostin 2015).

Kanava ja Innostimen nuorten ryhmä toteuttivat keväällä 2014 Duunari X -kampanjan, jonka tavoitteena oli kannustaa työnantajia palkkaamaan nuoria. Kampanjassa muun muassa järjestettiin työnantajien ja nuorten välille kohtaamis- ja verkostoitumistilaisuuksia ja tuotettiin YouTube-mainosvideo (Työ- ja elinkeinoministeriö 2014).

Duunari X -kampanjan osatuotteena Innostimen nuoret ideoivat nuorten elämänvaiheita ja Savonlinnan kaupungin tuki- ja apuverkosta mallintavan Nuoren kaupunki -WWW-sivuston ja kokosivat luettelon sivustoon tulevista palveluista. Sivuston tekninen suunnittelu ja toteutus ulkoistettiin kilpailutuksen kautta, koska olemassa olleesta organisaatiosta ei teknistä osaamista löytynyt. Tarjouspyynnön ja tarjouksen jälkeen tämän opinnäytetyön tekijä valittiin tehtävään maaliskuun 2014 lopulla.

Duunari X -ryhmän nuoret tuottivat sivuston tietosisällön ja karttakuvan. Opinnäytetyön tekijän vastuulla oli sivuston tekninen suunnittelu ja toteutus, tietosisällön jäsentäminen ja lisääminen sivustoon sekä käyttöönotto. Valmiin sivuston markkinoinnista vastaisi Kanava-hanke.

Yhteydenpito moniportaiseen asiakasorganisaatioon tapahtui maantieteellisestä etäisyydestä (Lappeenranta–Savonlinna) johtuen puhelimitse sekä sähköpostin ja Facebook-ryhmien välityksellä. Projektin alkuvaiheessa olin yhteydessä Nutun edustajaan käytännön asioissa, kuten opinnäytetyön lupa-anomuksen ja sivuston toimitussopimuksen osalta. Sivuston käyttöönottovaiheessa yhteydenpitoa vaati sivuston sijoituspaikaksi valittuun palvelimeen liittyneet asiat. Muutoin, erityisesti vaatimusmäärittelyn osalta, tein yhteistyötä pääasiassa Innostimen Duunari X -ryhmän projektipäällikön Petra Muhosen kanssa, joka välitti projektin edetessä myös muun organisaation ideoita ja korjausehdotuksia.

## **5.2 Esitutkimus ja määrittely**

Innostimen Duunari X -ryhmältä tulleen idean mukaan Nuoren kaupunki on WWW-sivusto, jota käytetään internetin kautta WWW-selaimella. Sivuston varsinaisena käyttöliittymänä on piirretty kaupungin kartta, jossa on nimettyjä paikkoja, kuten katuja, puistoja ja järviä. Paikat kuvaavat nuoren elämäntilanteita: vastavalmistunut nuori, työtön nuori, päihderiippuvainen nuori, ja niin edelleen. Paikkoja hiirellä napsauttamalla aukeaa luettelo elämäntilanteeseen liittyvistä palveluista Savonlinnan seudulla ja palvelua napsauttamalla tiedot kyseisestä palvelusta. Kuvassa 5.1 on Duunari X -ryhmältä tarjouspyynnön yhteydessä saamani luonnos kartasta.





Kuva 5.1. Luonnos kartasta.

Asiakkaan tärkein vaatimus oli, että sivuston tietosisällön tuli olla helposti päivitettävissä. Näin ollen sivuston taustalle luotaisiin tietokanta palvelutietojen tallentamista varten, ja sivustoon tulisi karttakäyttöliittymän lisäksi erillinen salasanasuojattu hallintakäyttöliittymä. Sivuston tuli toimia myös mobiililaitteissa. Alun perin keskusteltiin myös varsinaisen sivuston rinnalle kehitettävästä erillisestä mobiilisovelluksesta, mutta se rajattiin pois tavoittavuuden kyseenalaisuuden sekä tiukkojen aika- ja henkilöresurssien vuoksi.

Tärkein selvitettävä kysymys projektin alkuvaiheessa oli sivuston sijoituspaikka. Tarjouspyynnössä puhuttiin ylimalkaisesti ”Savonlinnan kaupungin kotisivuista”, mutta lopulta vaihtoehtoina olivat Nutun ja Innostimen sivustot tai kokonaan uusi webhotellipalvelu. Nutun sivusto (<http://www.nutu.fi/>) ei soveltunut kuvatuun kaltaiselle sivustolle siellä käytetyn Sporttisaiitti-julkaisujärjestelmän vuoksi. Järjestelmä sallii uusien sivujen lisäämisen vain tietyille esimuotoillulle sivupohjalle WWW-käyttöliittymän kautta. Kokonaan uusien WWW-sivujen lisääminen tai oman tietokannan liittäminen tähän järjestelmään ei ollut tekemäni selvityksen perusteella mahdollista. Innostimen sivusto (<http://www.innostin.fi/>) puolestaan sijaitsee fyysisesti Sigmatic Oy:n webhotellipalvelussa. Tämän selvittäminen

vaati tässä vaiheessa hieman luovuutta eli traceroute-työkalun käyttöä, millä jäljitin reitin innostin.fi-verkkotunnukseen. Sigmaticin webhotellissa on tuki muun muassa PHP-ohjelmointikielelle, MySQL-tietokannoille ja FTP-tiedonsiirrolle (Sigmatic 2015), joten sivusto päätettiin sijoittaa Innostimen sivuston alihakemistoon osoitteessa <http://www.innostin.fi/nuoreнкаupunki/>.

Sigmaticilla on tarjolla kolme erihintaista ja palveluvalikoimaltaan vaihtelevaa webhotellipalvelupakettia, Active, Advanced ja Premier (Sigmatic 2015). Innostimella oli käytössä paketeista suppein eli Active. Active-pakettiin ei sisälly MySQL-tietokantaa, joka oli hankittava lisäpalveluna. Active-pakettiin sisältyy myös vain yksi verkkotunnus, joten mikäli sivustolle haluttaisiin oma verkkotunnus, myös se olisi hankittava lisäpalveluna. Active-paketin liikennesuositus on viisi gigatavua kuukaudessa, jonka katsoin tässä vaiheessa riittävän sivuston tarpeisiin.

Määrittelyvaiheessa minulla ei vielä ollut pääsyä webhotellin palvelunhallintaan. Mahdollisten yhteensopivuusongelmien välttämiseksi otin yhteyttä Sigmaticin asiakaspalveluun liittyen käytössä olevien ohjelmistojen versioihin. Selvisi, että palvelimelle on asennettu PHP:stä versio 5.3.27 ja MySQL:stä versio 5.5. Käytössä olivat myös PHP:n MySQL Improved Extension (MySQLi)- ja MySQL Native Driver (Mysqli) -tietokantalaajennukset.

Sivuston kehityspalvelimena käytin omaan kotiverkkoon kytkettyä tietokonetta, johon oli asennettu nginx-WWW-palvelin, PHP (5.5.8), MySQL:ään pohjautuva MariaDB-tietokanta (5.5) ja phpMyAdmin-tietokannanhallintatyökalu. Palvelimen ohjelmistot oli asennettu jo kuukausia ennen opinnäytetyöprojektin alkamista, joten asennusprosessia ei kuvata tässä raportissa.

### **5.3 Sivuston osat**

Sivusto koostuu neljästä osasta, jotka ovat tietokanta, karttakäyttöliittymä, hallintakäyttöliittymä ja mobiilikäyttöliittymä. Aloitin osien toteuttamisen karttakäyttöliittymästä kuvassa 5.1 näkyvää karttaluonnosta pohjana käyttäen. Saatuani palvelutietoluettelon raakaversion suunnittelin kaiken taustalla toimivan tietokannan ja sitten hallintakäyttöliittymän, jonka jälkeen palasin viimeistelemään karttakäyttöliittymäliittymää. Lopuksi toteutin mobiilikäyttöliittymän.

Toteutuksessa käytin standardoituja HTML-kuvauskieltä ja CSS-tyyliohjeita. Palvelinpuolen ohjelmointikieleksi valikoitui PHP, joka oli tuttu jo aiemmista WWW-projekteista. Sigmaticin palvelimella on tuki myös Perlille ja Pythonille, joista ensimmäisestä minulla ei ollut kokemusta lainkaan ja jälkimmäisestäkin vain lyhyen verkkokurssin verran. Asiakaspuolella eli käyttäjän selaimessa tapahtuviin toimintoihin käytin avointa JavaScript-kirjastoa jQueryä.

En käyttänyt erityistä ohjelmointiympäristöä, vaan pääasiallisena koodieditorina toimi Notepad++-ohjelma.

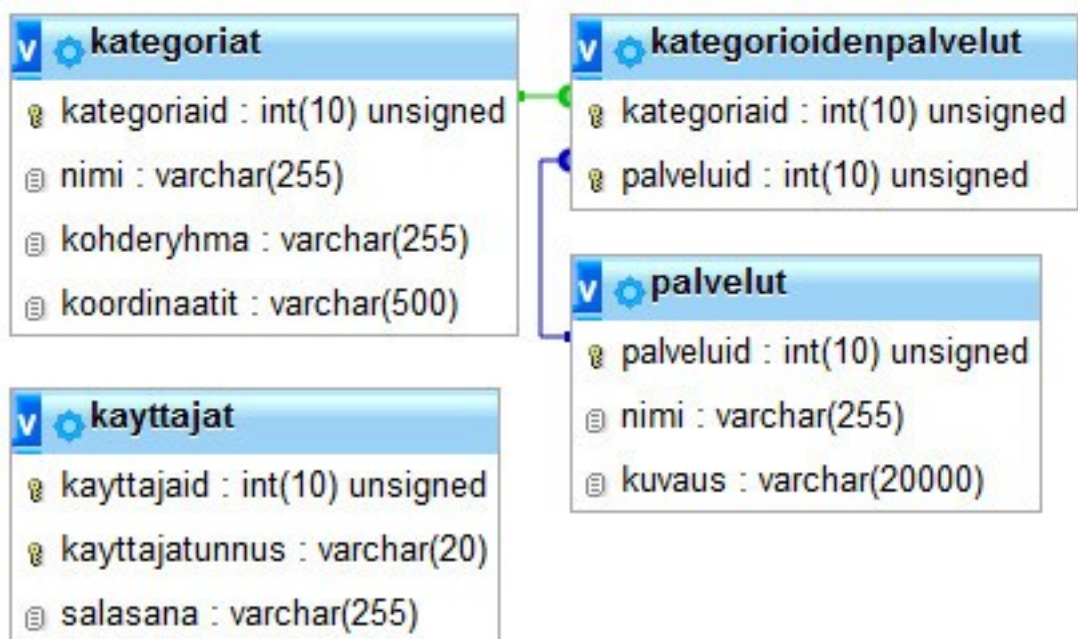
### **5.3.1 Tietokanta**

Sivuston taustalla toimii tietokanta, jonka suunnittelin Duunari X -ryhmältä saadun palveluluettelon pohjalta. Tietokannan toteutin MySQL-relaatiotietokantana, jonka valitsin tuttuutensa vuoksi ja koska se oli saatavilla sivuston tulevassa sijoituspaikassa Sigmaticin palvelimella. Käyttöönottovaiheessa kävi ilmi, että palvelimella olisi ilmeisesti tuki myös PostgreSQL-tietokannoille, mistä ei kuitenkaan ollut mainintaa palvelukuvauksessa (Sigmatic 2015).

Palveluluettelo oli DOC-muotoinen tekstitiedosto ja sisälsi palvelut kategorioitain (nuorten elämäntilanteet) listattuna. Kategorioita oli tässä vaiheessa 14 ja palveluita 187. Luettelo sisälsi runsaasti toistoa, sillä monessa tapauksessa täsmälleen sama palvelu kuului useampaan kuin yhteen kategoriaan ja oli listattu jokaisen kategorian alle erikseen. Ensitöiksi muutin luettelon logiikan päinvastaiseksi: kukin palvelu listattiin vain kerran, ja palvelun alle lisättiin maininta siihen liittyvistä kategorioista. Tämä myös helpotti palveluluettelon muokkaamista, kun haluttua muutosta ei tarvinnut tehdä moneen paikkaan, sekä aikanaan palveluiden lisäämistä tietokantaan, kun luettelossa oli mahdollista edetä palvelu kerrallaan miettimättä oliko palvelu jo mahdollisesti lisätty aiemmin. Palveluluettelon sisältö elikin koko työn ajan asiakasorganisaation tekemien lisäysten myötä. Lopulliseen sivustoon tuli 16 kategoriaa ja 78 palvelua.

Palvelutiedot itsessään koostuivat pääasiassa palvelun ja palveluun liittyvien henkilöiden yhteystiedoista. Tiedot olivat hyvin sekalaisia, sillä joihinkin palveluihin sisältyi useita paikkoja ja henkilöitä. Osassa oli sanallinen kuvaus palvelusta, mutta useimmissa ei. Tietoihin oli tarkoitus liittää myös yhteyshenkilöiden ku-

via ja Google Maps -katunäkymiä. Koska palveluille oli vaikea löytää yhteistä muotoa tietokannan rakenteen kannalta ja kunkin palvelun tiedot esitettäisiin sivustolla yhtenä kokonaisuutena, eikä esimerkiksi paikkoihin ja henkilöihin tarvitsi kohdistaa hakuja, päätin yksinkertaisuuden vuoksi tallentaa palvelun tiedot yhteen tauluun ja yhteen sisällöltään vapaamuotoiseen tekstikenttään useampaan kenttään ja tauluun hajauttamisen sijaan. Tekstikentän sisältö muotoiltiin hallintakäyttöliittymään tulevan editorin avulla. Näin tietokantaan tuli vain kaksi päätaulua, kategoriat ja palvelut. Jokaiseen kategoriaan kuului monta palvelua ja sama palvelu voi kuulua moneen kategoriaan. Kyseessä oli siis monen suhde moneen -yhteys, jota varten täytyi luoda välitaulu, joka yhdistää kategoriat ja palvelut toisiinsa. Sivustoon tuli myös hallintakäyttöliittymä, jonka käyttäjätillit tallennettiin tietokantaan omaan tauluunsa. Kuvassa 5.2 on tietokannan rakenne.

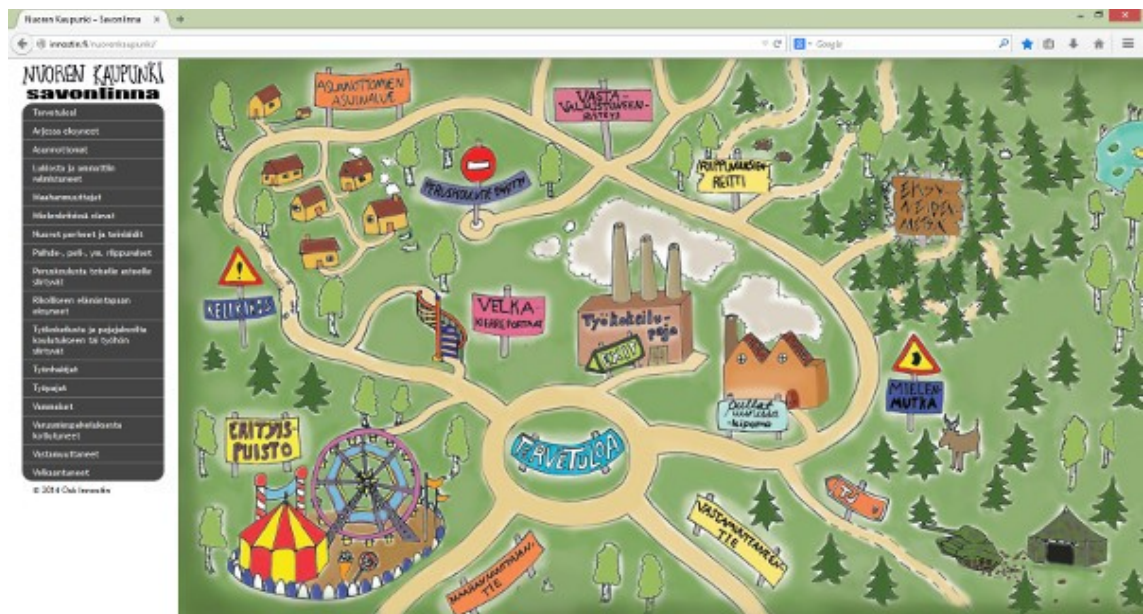


Kuva 5.2. Tietokannan rakenne.

Loin tietokannan ja sen taulut selainpohjaista phpMyAdmin-hallintaohjelmaa käyttäen. Tietokannan käsittelyyn itse sivustolla käytin PHP:n MySQLi-rajapintaa, joka on paranneltu versio alkuperäisestä MySQL-rajapinnasta. MySQL-rajapinta on julistettu vanhentuneeksi vuonna 2013 ilmestyneestä PHP:n versioista 5.5 alkaen, eikä sen käyttöä enää suositella (PHP Manual 2015a).

### 5.3.2 Karttakäyttöliittymä

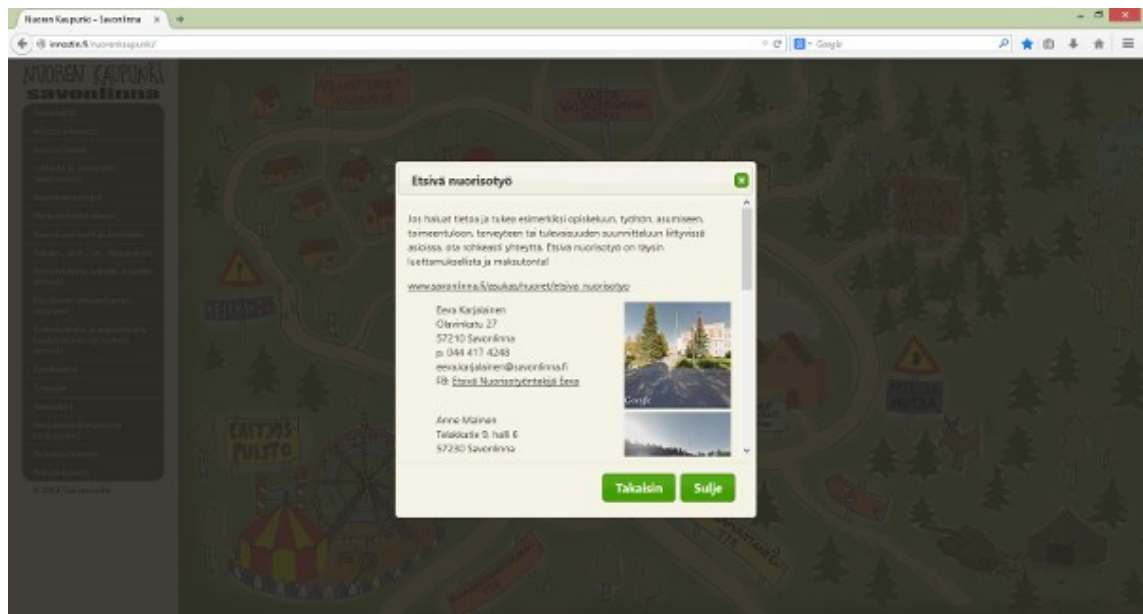
Sivuston näkyvin osa ja pääsivu on karttakäyttöliittymä, joka koostuu vasemman reunan valikosta ja loput selainikkunasta täyttävästä JPEG-muotoisesta piirretystä karttakuvasta. Kartalla on hiirellä napsautettavia nuoren elämäntilanteita kuvaavia paikkoja, ja valikko sisältää samat elämäntilanteet selkokielisempänä kohderyhmän nimellä. Käyttöliittymän toiminnot ja käyttötapaukset koostuvat siis yksinkertaisesti elämäntilannekategorioiden ja niihin liittyvien palveluiden selaamisesta. Kuvassa 5.3 on valmis karttakäyttöliittymä.



Kuva 5.3. Karttakäyttöliittymä.

Karttakuvan napsautettavat alueet toteutin HTML:n map- ja area-elementeillä. Napsautettavien alueiden koordinaattien määrittämiseen käytin Gimp-kuvankäsittelyohjelman kuvakarttaliitännäistä (Suotimet → Web → Kuvakartta). Muun muassa Stack Overflow -ohjelmointisivuston keskusteluissa pohditaan map-elementin käyttöä ja vaihtoehtoja sille (Stack Overflow 2014), mutta elementti on edelleen mukana W3C:n uusimmassa HTML-spesifikaatiossa HTML5:ssä (W3C 2014), eikä yksinkertaista ja yleispätevää vaihtoehtoa sille löytynyt. Haasteena oli saada kartta responsiiviseksi eli laajenemaan koko käytettävissä olevan selainikkunan levyiseksi ja korkuiseksi erikokoisilla näytöillä siten, että myös napsautettavat alueet pysyvät suhteessa oikeissa paikoissa. Tämä onnistui asiakaspuolella (käyttäjän selaimessa) jQuery:n avulla tehtävien laskutoimitusten avulla ja ImageMapster-liitännäisellä.

Valikon ja kartan ohella kolmas karttakäyttöliittymän pääelementti on ponnahdusikkunat, joihin kategorioiden ja palveluiden tiedot aukeavat. Valikko- ja karttalinkkien luonnin yhteydessä niille asetetaan id-attribuutti, joka saa saman arvon kuin kategorialla on tietokannassa. Käyttäjän napsauttaessa hiirellä valitsemaansa kategorialla avataan jQueryn Dialog-ponnahdusikkuna, johon ladataan jQueryn load()-metodin avulla HTML:llä ja PHP:llä toteutettu kategoriasivu. Saman load()-metodin kutsun yhteydessä tälle kategoriasivulle välitetään napsautetun elementin id-attribuutin arvo, jonka perusteella haetaan tietokannasta luettelo id:tä vastaavan kategorian palveluista. Load()-metodi on tyypiltään Ajax-metodi, joiden avulla palvelimelta voidaan ladata tietoa ilman selainikkunan päivittämistä (jQuery API 2015). Palveluiden avaaminen tapahtuu samaan tapaan kategorioiden kanssa. Kuvassa 5.4 on avattu palvelutietoikkuna.



Kuva 5.4. Erään palvelun tiedot jQueryn Dialog-ikkunassa.

jQueryä hyödyntävien käyttöliittymäelementtien ulkoasua on mahdollista muokata CSS:n avulla. jQueryn verkkosivustolla on joukko valmiita teemoja sekä ThemeRoller-työkalu omien teemojen luomista varten (jQuery UI 2015). Dialog-ponnahdusikkunoiden teemaksi valitsin valmiin South Street -teeman, jonka vihreät painikkeet sopivat yhteen karttakuvan kanssa.

Sivuston logon muokkasin Gimp-kuvankäsittelyohjelmassa. Logoa varten käytin kuvan 5.1 karttaluonnoksessa näkyvää Nuoren kaupunki -logoa sekä Tulikuumefonttia, johon hankin lisenssin dafont.com-sivustolta.

Karttakäyttöliittymää toteuttaessa otin huomioon sivuston mahdollisen laajentamisen uusilla kategorioilla tai jopa kokonaan eri kartan käyttämisen. Valikon sisältö ja kuvan napsautettavat alueet luodaan sivun latautuessa dynaamisesti tietokannasta haetuilla tiedoilla. Tämä mahdollistaa uusien kategorioiden lisäämisen muokkaamatta lähdekoodia.

### **5.3.3 Hallintakäyttöliittymä**

Hallintakäyttöliittymän kautta sivuston ylläpitäjä voi muokata sivuston kategorioiden ja palveluiden tietoja sekä vaihtaa ylläpitotunnuksen salasanan. Käytännössä hallintakäyttöliittymä on joukko WWW-sivuja ja -lomakkeita.

Hallintakäyttöliittymä sisältää seuraavat toiminnot ja käyttötapaukset:

- sisään- ja uloskirjautuminen
- kategorioiden hallinta
  - luettelo kategoriat
  - lisää kategoria
  - muokkaa kategoriaa
  - poista kategoria
- palveluiden hallinta
  - luettelo palvelut
  - lisää palvelu
  - muokkaa palvelua
  - poista palvelu
- käyttäjän hallinta
  - vaihda salasana.

Sivusto sijaitsee julkisella palvelimella, joten hallintakäyttöliittymä täytyi suojata käyttäjätunnuksella ja salasanalla. Koska HTTP on tilaton protokolla, toteutin hallintakäyttöliittymän sisään- ja uloskirjautumisen hallinnan sekä sivujen suojauksen PHP:n istuntofunktioiden ja -muuttujien avulla. Sisäänkirjautumissivulla käyttäjältä kysytään tietoja käyttöliittymän avatessaan, ja syötettyjä tietoja verra-

taan tietokantaan tallennettuihin tietojen. Onnistuneen sisäänkirjautumisen yhteydessä palvelimelle luodaan uusi käyttäjäkohtainen istunto. Jokaisella hallintakäyttöliittymän sivulla tarkastetaan onko istunto voimassa, ja mikäli ei ole, käyttäjä ohjataan sisäänkirjautumissivulle. Kuvassa 5.5 on hallintakäyttöliittymän etusivu onnistuneen sisäänkirjautumisen jälkeen.

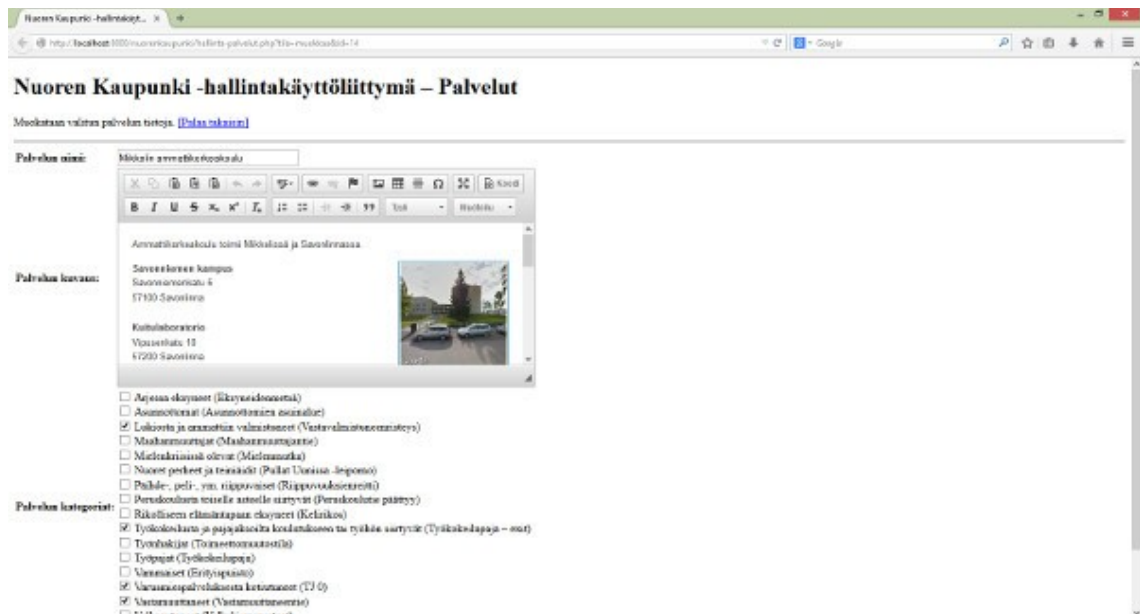


Kuva 5.5. Hallintakäyttöliittymän etusivu.

Kategorioiden hallinnassa syötettäviä tietoja ovat kategorian nimi, kohderyhmä ja koordinaatit kartalla. Näitä tietoja käytetään myös karttakäyttöliittymässä valikko- ja karttalinkkien luomiseen.

Palveluiden hallinnassa syötettäviä tietoja ovat palvelun nimi ja kuvaus sekä kategoriat, joihin palvelu kuuluu. Palvelun kuvausta muotoillaan HTML-tageilla WYSIWYG-editorin avulla. Tällaisia avoimen lähdekoodin editoreita on tarjolla lukuisia, ja sopivan löytäminen oli osoittautua hankalaksi. Muutaman yrityksen ja erehdyksen jälkeen editoriksi valikoitui CKEditor, joka vakuutti ominaisuuksillaan, selkeällä asetustiedostollaan ja helpolla käyttöönotollaan. Käyttöönotto tapahtuu käytännössä yhdellä JavaScript-rivillä, joka korvaa halutun textarea-elementin CKEditorilla. Kuvassa 5.6 on hallintakäyttöliittymän palvelun muokkauslomake.





Kuva 5.6. Palvelun muokkauslomake.

Palvelutietoihin kuului tekstin lisäksi myös yhteyshenkilöiden kasvokuvia ja Google Maps -katunäkymiä palveluihin liittyvistä osoitteista. Kasvokuvat tallennettiin fyysisesti sivuston alihakemistoon ja katunäkymäkuvat luotiin Google Street View Image API -rajapinnan avulla, minkä jälkeen kuvat linkitettiin editorissa osaksi palvelutietoja.

Käyttäjän hallinnassa on mahdollista vaihtaa ylläpitotunnuksen salasana. Tässä vaiheessa sivustolle tuli vain yksi ylläpitotunnus, mikä oli asiakkaan mukaan riittävästi. Tästä syystä käyttäjän hallintaan ei tullut muita ominaisuuksia, kuten uusien käyttäjien lisäämistä. Tietoturvan ja laajennettavuuden vuoksi en kuitenkaan sijoittanut käyttäjätietoja itse lähdekoodiin, vaan ne tallennettiin tietokantaan. Mahdollinen uusien käyttäjien lisääminen täytyisi tehdä esimerkiksi phpMyAdmin-hallintatyökalun kautta.

### 5.3.4 Mobiilikäyttöliittymä

Yksi sivuston vaatimuksista oli sen toimiminen myös mobiililaitteilla. Totesin karttakäyttöliittymän suuren karttakuvansa kanssa raskaaksi ja hankalaksi matkapuhelimen pienellä ruudulla käytettäväksi, joten toteutin matkapuhelimia varten erillisen mobiilikäyttöliittymän.

Mobiilikäyttöliittymä koostuu pelkästä tekstipohjaisesta valikosta, mutta on muuten toiminnoiltaan samanlainen karttakäyttöliittymän kanssa. Hyödynsin karttakäyttöliittymän yhteydessä luotua koodia, ja sivujen ulkoasuun käytin jQuery Mobile -kehystä.

Matkapuhelimella sivustolla vieraileva ohjataan automaattisesti mobiilikäyttöliittymään. Mobiilikäyttäjän tunnistamiseen käytin Mobile Detect -nimistä avoimen lähdekoodin PHP-skriptiä, joka tunnistaa käyttäjän selaimen lukemalla HTTP-kutsun otsikkotiedoista niin sanotun käyttäjäagentin (user agent) ja vertaa sitä skriptin mukana olevaan luetteloon tunnetuista mobiiliselaimista (Mobile Detect 2015). Skripti tunnistaa myös taulutietokoneet eli tabletit, joiden käyttäjät ohjataan karttakäyttöliittymään.

#### **5.4 Testaus**

Suoritin sivuston testauksen pääasiassa ohjelmoinnin lomassa toiminto kerrallaan. Todettuani kulloinkin työn alle olleen toiminnon toimivan halutulla tavalla siirryin seuraavaan toimintoon.

Ohjelmointivirheiden eliminointiin käytin PHP:n omaa virheraportointia, jonka asetin päälle kehityspalvelimen asetuksista, sekä Mozilla Firefox -selaimen web-kehittäjän työkaluja. Firefoxin työkalut sisältävät muun muassa HTML- ja CSS-tutkijan, virhekonsolin verkko-, CSS- ja JavaScript-virheille, JavaScript-debuggerin, jolla JavaScript-koodin suoritusta voi käydä läpi askel kerrallaan, ja verkkomonitorin, joka esittää selaimen sivustolle lähettämät pyynnöt aikajärjestyksessä. Kuvassa 5.7 on Firefoxin verkkomonitori toiminnassa.

Method	File	Domain	Type	Size	Time
304	GET	jquery.min.js	ajax.googleapis.com	js	93,24 KB
304	GET	jquery-ui.min.js	ajax.googleapis.com	js	223,10 KB
304	GET	jquery.imageMapster.min.js	innostin.fi	js	39,49 KB
304	GET	kartta.js	innostin.fi	js	5,63 KB
304	GET	login.png	innostin.fi	png	20,62 KB
304	GET	kartta.jpg	innostin.fi	jpeg	730,43 KB
200	POST	kartta-kategoria.php	innostin.fi	html	0,95 KB
304	GET	ui-bg_highlight-hard_100_156...	ajax.googleapis.com	png	0,50 KB
304	GET	ui-bg_gloss-wave_100_ece8da...	ajax.googleapis.com	png	7,55 KB
304	GET	ui-bg_highlight-hard_15_439a...	ajax.googleapis.com	png	0,49 KB
304	GET	ui-icons_eeeeee_256x240.png	ajax.googleapis.com	png	9,12 KB
200	POST	kartta-palvelu.php	innostin.fi	html	1,62 KB
200	GET	streetview?size=150x150&locat...	maps.googleapis.com	jpeg	6,68 KB
200	GET	streetview?size=150x150&locat...	maps.googleapis.com	jpeg	6,10 KB
304	GET	ui-bg_highlight-soft_25_67b0...	ajax.googleapis.com	png	0,52 KB
200	GET	ui-bg_highlight-hard_100_fafaf...	ajax.googleapis.com	png	0,48 KB
200	POST	kartta-palvelu.php	innostin.fi	html	0,60 KB

Kuva 5.7. Firefoxin verkkomonitori.

Sivuston HTML- ja CSS-koodin oikeellisuuden tarkastin W3C:n HTML- ja CSS-validaattoreilla. Sivusto läpäisi uusimmat HTML5- ja CSS3-määrittymät.

Sivuston ulkoasulle suoritin pintapuolisen vertailun Microsoft Windows -käyttöjärjestelmän yleisimpien selainten Google Chromen (versio 35), Mozilla Firefoxin (30), Internet Explorerin (11) ja Operan (22) uusimmilla versioilla ja varmistin näin, että sivusto näyttää samalta selaimesta riippumatta. Internet Explorerin osalta, joka päivittyy harvemmin ja noudattaa huonosti standardeja erityisesti vanhempien versioidensa osalta (HTML5test 2015), tein vertailun myös versioilla 8–10. Vanhemmat versiot olivat ladattavissa virtuaalikoneina Microsoftin Modern.IE-sivustolta.

Mobiilikäyttöliittymän toiminnan testasin sekä Android-älypuhelimella että -taulutietokoneella Google Chrome-, Mozilla Firefox- ja Opera-selaimilla.

## 5.5 Dokumentointi

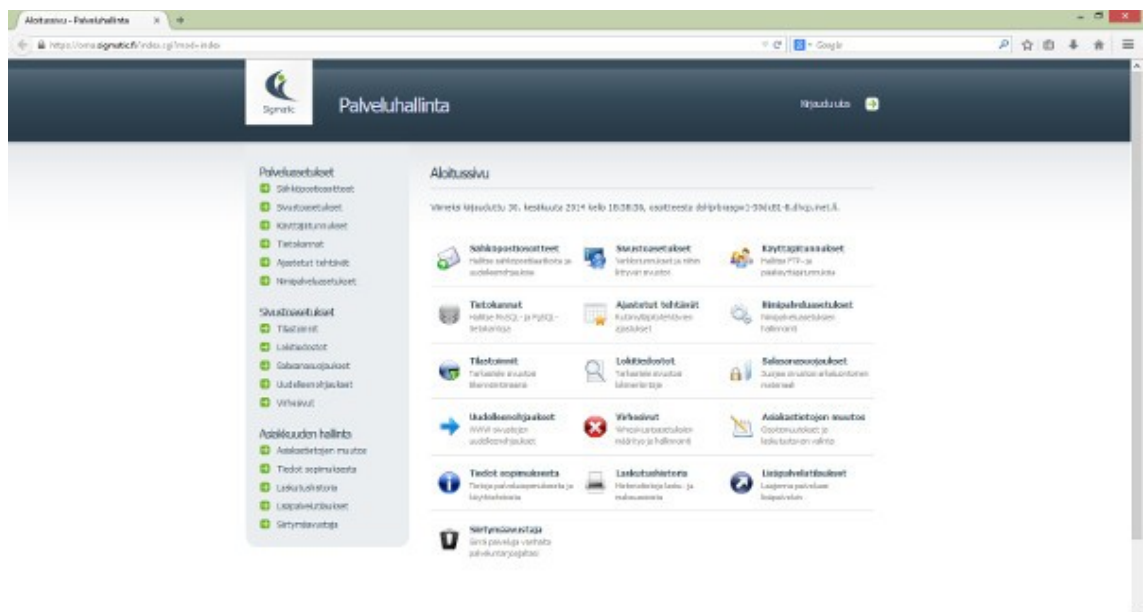
Sivuston PHP-, HTML-, CSS- ja JavaScript (jQuery) -koodi sisältää lyhyet kommentit eri tiedostojen ja koodin osien käyttötarkoituksesta.

Lähdekoodin lisäksi toimitin asiakkaalle DOC-muotoiset ohjeet sivuston teknisistä vaatimuksista ja asentamisesta sekä palvelujen päivittämisestä.

## 5.6 Käyttöönotto

Ennen sivuston käyttöönottoa lisäsin kehityspalvelimella palvelutiedot hallintakäyttöliittymän kautta tietokantaan. Tämä tapahtui palvelu kerrallaan kopioimalla tiedot alkuperäisestä DOC-dokumentista ja muotoilemalla ne hallintakäyttöliittymän editorissa.

Saatuani asiakkaalta tunnukset Innostimen webhotelliin Sigmatic Oy:n palvelimella ja valtuudet tehdä tarvittavat muutokset webhotellipalvelupakettiin alkoi varsinainen käyttöönotto. Palvelupakettia hallitaan WWW-selaimessa toimivan hallintapaneelin kautta. Koska Innostimen palvelupakettiin ei kuulunut MySQL-tietokantaa, täytyi se hankkia lisäpalveluna. Tämä tapahtui hallintapaneelin Lisäpalvelutilaukset-sivun Laajennukset-välilehden kautta. Tilauksen tekemisen ja laskun maksamisen jälkeen loin tietokannan ja tietokantakäyttäjän hallintapaneelin Tietokannat-sivun kautta. Kuvassa 5.8 on webhotellin hallintapaneelin etusivu.



Kuva 5.8. Webhotellin hallintapaneelin etusivu.

Muita muutoksia palvelupakettiin ei tarvinnut tehdä, sillä asiakkaan mukaan sivustolle ei ollut tarpeellista rekisteröidä omaa verkkotunnusta. Myöskään palvelupaketin viiden gigatavun kuukausittaista liikennöintisuositusta ei ollut tarvetta nostaa, sillä hallintapaneelin tilastotyökalujen mukaan alkuvuoden 2014 aikana

Innostimen sivustolta oli lähtenyt liikennettä vain keskimäärin noin 15 megatavua kuukaudessa.

Käyttöönottoon kuului myös tietokannan ja tiedostojen siirtäminen kehityspalvelimelta Sigmaticin palvelimelle. Tietokannan siirtämiseen käytin phpMyAdmin-tietokannanhallintatyökalua. Tein aluksi kehityspalvelimella vedoksen tietokannasta phpMyAdminin vienti-toiminnon avulla. Toiminto luo tekstitiedoston, joka sisältää tietokannan taulujen rakenteen ja sisällön valmiina SQL-lauseina (CREATE TABLE ja INSERT). Tämän jälkeen siirsin taulut Sigmaticin palvelimelle phpMyAdminin tuonti-toiminnolla.

Tiedostojen siirtämiseen käytin FileZilla-ohjelmaa. Otin ohjelmalla SFTP-yhteyden Sigmaticin palvelimelle Sigmaticin verkkosivustolta löytyvien ohjeiden mukaisesti ja siirsin tiedostot public\_html-hakemistoon luomaani nuorenkaupunkialihakemistoon. Käyttöönoton viimeisenä toimenpiteenä oli asettaa tiedostoille oikeanlaiset luku-, kirjoitus- ja suoritusoikeudet ohjeiden mukaisesti (chmod 644). Lopuksi siirryin WWW-selaimella verkko-osoitteeseen <http://www.innostin.fi/nuorenkaupunki/>, josta sivusto vastasi ongelmitta. Nuoren kaupunki -sivusto oli näin otettu käyttöön.

Sivuston virallinen lanseeraustilaisuus järjestettiin Savonlinnassa 30. kesäkuuta 2014 (Jääskeläinen, P. 2014). Sivuston markkinoinnista vastasi Kanava-hanke, joka julkaisi aiheesta lehdistötiedotteen. Sivustoa markkinoitiin Nutun ja Savonlinnan kaupungin sivustojen lisäksi ainakin Po1nt – nuorten portaalissa, Nuorisotakuu-sivustolla ja ELY-keskuksen sivustolla sekä Facebookissa eri sivuilla.

## **6 Pohdinta**

Opinnäytetyönä suunniteltiin ja toteutettiin Savonlinnan Seudun Nuorisotoiminnan Tuki ry:lle ja Savonlinnan kaupungin Kanava – työvoimaa avoimille työmarkkinoille -hankkeelle nuoren elämäntilanteita sekä Savonlinnan kaupungin tuki- ja apuverkostoa esittelevä WWW-sivusto.

Sivusto valmistui aikataulussa ja sisälsi asiakkaiden vaatimat ominaisuudet, joten työn voi näiltä osin katsoa täyttäneen tavoitteensa. Asiakkaat olivat lopputu-

lokseen tyytyväisiä, ja sivusto ja idea herättivät kiinnostusta myös Etelä-Savon elinkeino-, liikenne- ja ympäristökeskuksessa. Olen myös itse tyytyväinen aikaansaannoksiini kenties sivuston hallintakäyttöliittymän hyvin pelkistetyksi jäänyttä ulkoasua lukuun ottamatta.

Ennen opinnäytetyötä opintoni olivat olleet tauolla parin vuoden ajan, mikä ilmeni lievänä suunnittelemattomuutena ja poukkoiluna. Tarkempi suunnittelu ja etenkin työn vaiheiden dokumentointi olisi ehkä ollut tarpeen, sillä nyt asioita tuli tehtyä paljolti niin sanotusti lennossa ja omassa päässä. Toisaalta, kun sivuston tekninen toteutus oli vain yhden hengen projekti, pysyivät langat näinkin käsissä ja lopputuloksena oli toimiva kokonaisuus. Vähäinen dokumentointi hankaloitti kuitenkin opinnäytetyön raportointivaihetta, joka kesti huomattavan kauan.

Opinnäytetyössä käytetyt tekniikat olivat enimmäkseen tuttuja aiemmista WWW-projekteista. Myös näistä oli kuitenkin kulunut aikaa, joten opinnäyte palautti hyvin mieleen PHP- ja SQL-kielet ja harjaannutti niiden käytössä. Uutena asiana tuli PHP:n MySQLi-tietokantarajapinta ja siihen liittyen MySQL:n valmistellut lausekkeet, sillä aiemmin olin käyttänyt vain vanhempaa, tietoturvaltaan heikompaa MySQL-rajapintaa. Toinen uusi asia oli valmis JavaScript-kirjasto jQuery, joka osoittautui hyväksi ja helppokäyttöiseksi välineeksi käsitellä HTML-sivun elementtejä ja tapahtumia ja tuottaa sivuille dynaamista sisältöä. Uskon molemmista olevan jatkossa paljon hyötyä.

Jälkeenpäin ajateltuna hieman harmittaa, etten tutustunut valmiiden sisällönhallintajärjestelmien, kuten Wordpressin, Joomla:n tai Drupalin, suomiin mahdollisuuksiin tämänkaltaisessa projektissa. Voi toki olla, että niiden käyttäminen sivuston pohjana olisi vaatinut runsaasti muokkaamista ja lopputulos olisi sisältänyt tarpeettomia komponentteja. Nyt käytetyllä omalla ratkaisulla kokonaisuus pysyi hyvin rajattuna.

## Kuvat

Kuva 2.1. Esimerkki tietokannan tauluista, s. 11

Kuva 2.2. CKEditor, s. 14

Kuva 3.1. PhpMyAdminin käyttöliittymä, s. 16

Kuva 4.1. Suojatun yhteyden kuvake ja tiedot, s. 21

Kuva 5.1. Luonnos kartasta, s. 25

Kuva 5.2. Tietokannan rakenne, s. 28

Kuva 5.3. Karttakäyttöliittymä, s. 29

Kuva 5.4. Erään palvelun tiedot jQueryn Dialog-ikkunassa, s. 30

Kuva 5.5. Hallintakäyttöliittymän etusivu, s. 32

Kuva 5.6. Palvelun muokkauslomake, s. 33

Kuva 5.7. Firefoxin verkkomonitori, s. 35

Kuva 5.8. Webhotellin hallintapaneelin etusivu, s. 36

## Lähteet

CKEditor 2015. About. <http://ckeditor.com/about> Luettu 30.5.2015.

Gerber, R. & Compton, T. 2014. 10 security tips to protect your website from hackers. Creative Bloq. <http://www.creativebloq.com/web-design/website-security-tips-protect-your-site-7122853> Luettu 30.5.2015.

Google Developers 2015. Google Street View Image API. <https://developers.google.com/maps/documentation/streetview/> Luettu 30.5.2015.

Haikala, I. & Märijärvi, J. 2002. Ohjelmistotuotanto. Helsinki: Talentum.

Heinisuo, R. & Rauta, I. 2007. PHP ja MySQL – Tietokantapohjaiset verkkopalvelut. Helsinki: Talentum.

HTML5test 2015. How well does your browser support HTML5? <http://html5test.com/results/desktop.html> Luettu 30.5.2015.

Innostin 2015. Mitä? <http://www.innostin.fi/mita.php> Luettu 30.5.2015.

ImageMapster 2015. What Is It. <http://www.outsharked.com/imagemapster/default.aspx?what.html> Luettu 30.5.2015.

jQuery API 2015. <http://api.jquery.com/> Luettu 30.5.2015.

jQuery UI 2015. ThemeRoller. <http://jqueryui.com/themeroller/> Luettu 30.5.2015.

Jääskeläinen, P. 2014. Innostimen nuoret suunnittelivat uuden sivuston nuorille. Itä-Savo 1.7.2014, 3.

Mobile Detect 2015. <http://mobiledetect.net/> Luettu 30.5.2015.

Nutu 2015. Tietoa Nutusta. <http://www.nutu.fi/tietoa-nutusta/> Luettu 30.5.2015.

Paananen, J. 2001. Tietotekniikan peruskirja. Jyväskylä: Docendo.

PHP Manual 2015a. Choosing an API. <http://php.net/manual/en/mysqlinfo.api.choosing.php> Luettu 30.5.2015.

PHP Manual 2015b. Prepared Statements. <http://php.net/manual/en/mysqli.quickstart.prepared-statements.php> Luettu 29.5.2015.

PhpMyAdmin 2015. Features. [http://www.phpmyadmin.net/home\\_page/](http://www.phpmyadmin.net/home_page/) Luettu 30.5.2015.



Savonlinna 2015a. Kanava – työvoimaa avoimille työmarkkinoille.  
<http://www.savonlinna.fi/savonlinna-tietoa/kehittamishankkeet/kanava-projekti>  
Luettu 30.5.2015.

Savonlinna 2015b. Neppari – nuorten asian yhteisöryhmä.  
<http://www.savonlinna.fi/asukas/nuoret/neppari> Luettu 30.5.2015.

Sigmatic 2015. Webhotellit. <http://www.sigmatic.fi/palvelut/webhotellit/> Luettu 30.5.2015.

Stack Overflow 2015. Posts containing 'image map'.  
<http://stackoverflow.com/search?q=image+map> Luettu 30.5.2015.

Työ- ja elinkeinoministeriö 2014. Duunari X tuo nuoria ja työnantajia yhteen Savonlinnan seudulla.  
[http://www.tem.fi/nuorisotakuu/tietoa\\_takuusta/ajankohtaista/duunari\\_x\\_tuo\\_nuoria\\_ja\\_tyonantajia\\_yhteen\\_savonlinnan\\_seudulla.114023.news](http://www.tem.fi/nuorisotakuu/tietoa_takuusta/ajankohtaista/duunari_x_tuo_nuoria_ja_tyonantajia_yhteen_savonlinnan_seudulla.114023.news) Luettu 30.5.2015.

Ullman, C. 2007. What Is Ajax? <http://www.wrox.com/WileyCDA/Section/id-303217.html> Luettu 30.5.2015.

W3C 2014. HTML 5. <http://www.w3.org/TR/html5/> Luettu 30.5.2015.

W3C 2015. HTML & CSS. <http://www.w3.org/standards/webdesign/htmlcss.html>  
Luettu 30.5.2015.

W3Techs 2015. Usage of server-side programming languages for websites.  
[http://w3techs.com/technologies/overview/programming\\_language/all](http://w3techs.com/technologies/overview/programming_language/all) Luettu 30.5.2015.