

Opinnäytetyö (AMK)

Tietotekniikka

Mediatekniikka

2014

Oskari Heinonen

FRONT END -KEHITYSTYÖ- KALUJEN HYÖDYNTÄMINEN VERKKOSIVUSTON KEHITYKSESSÄ

- case: lolleipuri.fi



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

Oskari Heinonen

FRONT END -KEHITYSTYÖKALUJEN HYÖDYNTÄMINEN VERKKOSIVUSTON KEHITYKSESSÄ

Opinnäytetyössä tutkitaan front end -kehitystyökalujen hyödyntämistä verkkosivujen kehityksessä. Opinnäytetyö suoritettiin verkkosivustoprojektina Martinex Oy:lle, jolle kehitettiin reseptinhakusivusto Iloleipuri-tuoteperheelle.

Front end -kehityksellä tarkoitetaan asiakkaan puolista verkkokehittämistä. Yksi verkkokehittäjän tärkeimmistä työkaluista on tekstieditori. Ensimmäisessä osiossa käytiin läpi yleisimmät tekstieditorit, jossa vielä tarkemmin syvennyttiin Sublime Text -editorin toiminnallisuuksiin ja muokattavuuteen. Front end -kehittäjälle on tarjolla myös lukuisa määrä avoimen lähdekoodin kirjastoja, kehyksiä ja mallipohjia. Näitä käyttämällä verkkokehitys tehostuu. Sisällönhallintajärjestelmä sivuston taustalla on hyvä vaihtoehto projekteissa, joissa asiakas vastaa sisällön päivityksestä itse. Opinnäytetyössä tarkasteltiin myös selaimia osana verkkokehityksen työnkulkua ja tutustuttiin verkosta löytyviin resursseihin, kuten isännöintipalveluihin ja verkko-ohjelmointiympäristöihin. Lopuksi todettiin versionhallinnan hyödyllisyys verkkosivuston kehityksessä. Sen avulla useampi tiimin jäsen pystyi kehittämään verkkosivustoa tehokkaasti yhdessä.

Tutkimusten pohjalta valittiin front end -työkalut ja -tekniikat Iloleipuri.fi-sivuston rakentamista varten. Tekstieditoriksi ladattiin Sublime Text ja pääasialliseksi kehitysselaimeksi Google Chrome. Sass valittiin CSS-esikäsittelijäksi, ja sivusto rakennettiin Wordpress-julkaisujärjestelmän päälle. Wordpressin kautta ladattiin etusivun kuvakaruselille oma lisäosa. Sivustolle otettiin käyttöön myös JavaScript-pohjaisia lisäosia, kuten esimerkiksi imageScroll.js. Git valittiin versionhallintaohjelmaksi ja sitä käytettiin PowerShell-komentotulkilla.

Erlaisia työkaluja on rakennettu lähes jokaiselle front end -verkkokehityksen osa-alueelle. Tärkeintä ei kuitenkaan ole ottaa käyttöön kaikkia uusi työkaluja vaan oleellisinta on valikoida niistä itselleen sopivat.

ASIASANAT:

verkko-ohjelmointi, verkkokehittäjä, front end, tekstieditori, kehys, sisällönhallintajärjestelmä, versionhallinta, iloleipuri

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information technology | Digital Media

2014 | 44

Mika Luimula, Principal Lecturer, PhD

Oskari Heinonen

UTILIZING FRONT-END DEVELOPMENT TOOLS IN WEBSITE DEVELOPMENT

The purpose of this thesis was to examine the usage of the front-end development tools in website development. The thesis was carried out as a website project for Martinex Oy. A recipe search site was created for the Iloleipuri product family during the project.

Front-end development means the client side web development. One of the most important tools used by web developers is a text editor. The first part of the thesis examines the most common text editors. The Sublime Text editor and its functionalities are discussed in detail. There is a great number of open source libraries and frameworks. The use of these makes developing a website more efficient. Installing content management system in the background of the website is a good option for projects where the customer is responsible for the content updates. The thesis also discusses the browser as part of the web development workflow and resources found from the Internet, such as hosting services and online JavaScript integrated development environments. Lastly, the thesis evaluates how beneficial it was to create websites with a version control system. A version control system allows several team members to develop a website efficiently together.

Front-end development tools and techniques for developing Iloleipuri.fi were selected based on the above mentioned research. For the text editor, Sublime Text was downloaded and the main development browser was chosen to be Google Chrome. Sass was selected as a CSS preprocessor and the site was built on Wordpress. The plugin for Wordpress was downloaded to manage the image carousel of the front page. The website also used JavaScript-based plugins, such as imageScroll.js. Git version control system was selected and used with a PowerShell command line tool.

Different types of tools are built for almost every area of the front-end web development. However, it is important not to start using every new tool but to select suitable tools for a project.

KEYWORDS:

web programming, web developer, front end, text editor, framework, content management system, version control system, iloleipuri

SISÄLTÖ

SANASTO	6
1 JOHDANTO	8
2 FRONT END -KEHITYS	10
3 TEKSTIEDITORIT	12
4 KEHYKSET, KIRJASTOT SEKÄ CSS-ESIKÄSITTELIJÄT	17
4.1 Kehykset	18
4.2 Kirjastot	19
4.3 CSS-esikäsittelijät	20
5 SISÄLLÖNHALLINTAJÄRJESTELMÄ VERKKOSIVUN TAUSTALLA	24
6 SELAIMET JA VERKKOTYÖKALUT	27
6.1 Kehittäjän työkalut selaimissa	27
6.2 Verkkotyökalut	28
7 VERSIONHALLINNAN KÄYTTÖ OSANA VERKKOKEHITYSTÄ	30
8 ILOLEIPURI.FI-SIVUSTON RAKENTAMISEN ESIVALMISTELUT	32
9 ILOLEIPURI.FI-SIVUSTON TOTEUTUS	34
9.1 Työnkulku ja versionhallinnan käyttö projektiryhmässä	34
9.2 Käytetyn CSS-esikäsittelijän hyödyntäminen	34
9.3 Kehittäjän työkalut selaimessa osana testausta	35
9.4 Valittujen kirjastojen ja JavaScript-pohjaisten lisäosien käyttöönotto	36
9.5 Wordpressin käyttö sivuston sisällönhallintajärjestelmänä	37
9.6 Sivuston toteutus ja julkaisu	38
10 POHDINTA	40
LÄHTEET	42

KUVAT

Kuva 1 Sublime Text -tekstieditorin käyttöliittymä	14
Kuva 2 Git-versionhallinnan käyttö Windows PowerShell -komentotulkilla	30
Kuva 3 Sivuston ruudunpäivityksen parantaminen kehitystekniikoita vaihtamalla	36
Kuva 4 Iloleipuri-sivuston etusivu	39

TAULUKOT

Taulukko 1 Tekstieditorien vertailua	13
Taulukko 2 Vertailua yleisimpien sisällönhallintajärjestelmien välillä	25

SANASTO

Avoin lähdekoodi	Ohjelmien yhteisöllinen tuottamis- ja kehitysmenetelmä. (engl. Open Source)
Back end	Kehityksen osa-alue, joka yleisesti käsittää palvelimella suoritettavan ohjelmoinnin.
Brackets	Adobeen kehittämä avoimen lähdekoodin tekstieditori, jonka ylläpito tapahtuu nykyään GitHubin kautta.
CSS	Verkkosivujen ulkoasun muokkausta varten kehitetty tyyliohjeiden laji. (Cascading Style Sheets)
Dokumenttioliomalli	Ohjelmointirajapinta, joka mahdollistaa HTML-tiedoston elementtien muokkaamisen. (engl. DOM, document object model)
Eväste	Käyttäjän tietokoneelle serveriltä tallennettua tietoa. (engl. Cookie)
Front end	Kehityksen osa-alue, joka kattaa käyttäjänpuolisen kehittämisen.
FTP	Tiedonsiirtomenetelmä kahden tietokoneen välille. (File Transfer Protocol)
GitHub	Verkkopohjainen isännöintipalvelu projekteille, jotka käyttävät Git-versionhallintaa.
Grunt	JavaScript tehtävän suorittaja.
HTML	Merkintäkieli verkkosivujen kehittämiseen. (Hypertext Markup Language)
HTTP	Protokolla, jota selaimet käyttävät tiedonsiirtoon. (engl. HTTP, Hypertext Transfer Protocol Request)
JavaScript	Komentosarjakieli, jota käytetään verkkoympäristössä.
Kehys	Tuote, joka tarjoaa kehitykselle rungon. (engl. Framework)
Kirjasto	Kokoelma, jonka tarjoamia luokkia ja ohjelmia voidaan käyttää hyväksi kehityksessä. (engl. Library)
Kuvakaruselli	Elementti verkkosivulla, jossa vaihtuu säännöllisin väliajoin ennalta määrätyt kuvat. (engl. Image Carousel)
Mallipohja	Valmis pohja, joka nopeuttaa tuotteen kehitystä. (engl. Boilerplate)
Node.js	Skaalautuva JavaScriptillä toteutettu ohjelmistoalusta serverillä ajettaville ohjelmille.

Parallaksi-efekti	Efekti, joka tapahtuu, kun havaitsija ja havaittavat kohteet liikkuvat eri nopeuksilla. (engl. Parallax)
PHP	Palvelinpuolen ohjelmointikieli. (Hypertext Preprocessor)
Responsiivinen	Responsiivinen verkkosivu mukauttaa sivuston sisällön näyttöpäätteen koon mukaan. (engl. responsive)
Ruudunpäivitys	Luku, joka kertoo kuinka monta kertaa sekunnissa näyttöruutua päivitetään. (engl. FPS, Frames Per Second)
Sass	CSS-esikäsittelijä, joka laajentaa normaalin CSS:n ominaisuuksia.
Selainkohtainen etuliite	CSS-tyylimääritys uusien CSS-ominaisuuksien tuelle. (engl. Vendor Prefix)
Sisäkkäiset säännöt	CSS-esikäsittelijöiden mahdollistava tekniikka, jossa CSS-valitsimia voidaan kirjoittaa myös sisäkkäin. (engl. Nested Rule)
SQL	Kyselykieli, jolla voi tehdä hakuja relaaliotietokantoihin. (Structured Query Language)
Tehtävän suorittaja	Ohjelma, joka suorittaa automaattisesti joukon toimintoja kehityksen taustalla. (engl. Task Runner)
Typografia	Tekniikka kirjainten ja tekstin asettelulle ja tyyleille.
Työnkulku	Prosessi, jonka kehittäjä toistaa kehittäessään tuotetta. (engl. Workflow)
Valitsin	CSS:ssä käytetty DOM-elementin nimi, jolle tyylimääritykset asetetaan. (engl. Selector)
Varasto	Versionhallinnassa käytetty tiedontallentamiskohde. (engl. Repository)
Vastauksen ylätunniste	Tekstimuodossa olevaa tietoa, jonka serveri lähettää takaisin käyttäjän tietokoneelle. (engl. Response Headers)
Verkoapplikaatio	Sovellusohjelma, joka ajetaan verkkoselaimessa. (engl. Web-Application)

1 JOHDANTO

Opinnäytetyössä tutkitaan, miten front end -kehityksessä käytettävät työkalut ovat kehittyneet viimeisten vuosien aikana, ja miten se on vaikuttanut työnkulkuun (engl. workflow). Samalla kartoitetaan, mitä työkaluja opinnäytetyönä kehitettävän Iloleipuri-reseptinhakusivuston rakentamiseen valittiin ja kuinka niitä hyödynnettiin verkkosivuston kehityksen aikana.

Asiakkaanpuolinen (engl. client side) verkkokehittäminen eli front end -kehitys ja siinä käytettävät työkalut kehittyvät jatkuvasti eteenpäin. Nopean kehityksen taustalla on aktiivinen verkkokehittäjien yhteisö. Uusia kehyksiä (engl. framework), kirjastoja (engl. library) ja työkaluja rakennetaan jatkuvasti lisää yhteisön tuen avulla. Tässä opinnäytetyössä tuodaan esille tämän päivän käytetyimpiä ja hyödyllisimpiä työkaluja sekä suoritetaan vertailua niiden välillä.

Ensimmäisenä tarkastelun kohteena on verkkokehittäjän tärkein väline eli tekstieditori. Tekstieditoreita on lukuisia erilaisia, mutta tässä opinnäytetyössä niistä käsitellään yleisimmät ja tutkitaan niitä front end -kehittäjän näkökulmasta. Sublime Text -nimiseen tekstieditoriin syvennytään tarkemmin.

Front end -kehittäjälle on tarjolla lukuisa määrä avoimen lähdekoodin (engl. open source) kirjastoja, kehyksiä ja mallipohjia (engl. boilerplate). Käyttämällä näitä oikein verkkokehitys on tehokasta ja nopeaa. Opinnäytetyössä käsitellään näistä yleisimpiä ja tutkitaan, millä tavalla kukin parantaa kehittäjän arkea. CSS-esikäsittelijöistä syvennytään tarkemmin Sass-käsittelijän ominaisuuksiin. Verkkosivusto on myös mahdollista rakentaa sisällönhallintajärjestelmän päälle. Seuraavassa osiossa tutkitaan näitä ja syvennytään tarkemmin Wordpressin käyttöön. Tämän lisäksi tutkitaan verkkoselaimia osana front end -kehittäjän työnkulkua. Kaikki yleisimmät verkkoselaimet pitävät sisällään kehittäjän työkalut (engl. developer tools), jotka mahdollistavat sivustojen reaaliaikaisen kehittämisen. Tässä yhteydessä tutkitaan myös selaimella käytettäviä verkkotyökaluja ja -sivuja, joita kehittäjä voi tarvitessaan käyttää verkkosivuston rakennuksessa.

Versionhallintaohjelma on tärkeä työkalu, kun kehitystiimissä on useampi jäsen. Versionhallintaohjelman avulla tiedostojen jakaminen helpottuu ja niiden varmuuskopiointi automatisoituu. Opinnäytetyössä on osio, jossa otetaan selvää, millä tavalla niitä tulee käyttää, jotta isommankin tiimin kehitysprosessi on selvästi sujuvampaa.

Opinnäytetyön tarkoitus oli kartoittaa oikeat työvälineet Iloleipuri-reseptinhakusivuston rakentamiselle. Tutkimusten pohjalta, valittiin käytettävät kehitystyökalut. Sivusto rakennettiin neljän hengen kehitystiimillä.

Opinnäytetyön projektiosuus aloitetaan käymällä läpi toimeksianto, projektin esivalmistelut ja käyttöönotettavat työtavat. Tämän jälkeen siirrytään projektin toteutukseen. Tässä keskitytään Iloleipuri-sivuston rakentamisessa käytettyihin front end -tekniikoihin ja -työkaluihin. Näiden lisäksi syvennytään yksittäisen tiimin jäsenen työkuluun ja tutkitaan, kuinka erilaisten front end -kehitystyökalujen käyttö on parantanut sitä.

2 FRONT END -KEHITYS

Front end -kehityksellä tarkoitetaan käyttäjänpuolista verkkokehittämistä eli verkkosivuston käyttäjälle näkyvien toiminnallisuuksien ja elementtien rakentamista. Front end -kehittäjän tulee tänä päivänä hallita hyvän käyttäjäkokemuksen (engl. user experience/UX) luominen. Käyttäjäkokemus kattaa kaikki näkökulmat käyttäjän vuorovaikutuksesta yrityksen, sen palveluiden ja tuotteiden kanssa. Hyvä käyttäjäkokemus syntyy, kun verkkosivusto tai -sovellus täyttää käyttäjän toiveet ilman turhaa vaivaa, ja sen käyttöliittymä (engl. user interface) on selkeä ja helppokäyttöinen. Myös palvelun käytettävyydellä on iso rooli hyvän käyttäjäkokemuksen luomisessa. Käytettävyys on käyttöliittymän laatumääre, joka kattaa palvelun käytön helppouden, tehokkuuden ja oppimisen. Käytettävyydeltään hyvän verkkosivun tarjoamat palvelut yhdistyvät saumattomasti sivuston tai sovelluksen hallitusti suunniteltuun ulkoasuun ja käyttöliittymään. [1,2]

Front end -kehitys mielletään usein liittyvän vain verkkosivustojen ulkonäöllisiin ominaisuuksiin, mutta front end -kehitys käsittää kuitenkin tekniikkaa myös käyttöliittymän takaa. Merkintäkieli HTML:n (Hypertext Markup Language) ja tyylistivukieli CSS:n (Cascading Style Sheets) lisäksi front end -kehitykseen kuuluu myös oliopohjainen komentosarjakieli JavaScript. Sen avulla kehittäjän on mahdollista esimerkiksi manipuloida HTML-elementtejä käyttäjän selaimessa. Käyttöliittymän muokkaamisen lisäksi, JavaScriptillä onnistuu myös tänä päivänä yhteyksien rakentamiset käyttöliittymän ja tietokannan välillä. JavaScriptillä on näin siis rikottu front end - ja back end -kehityksen rajaa. Back end -kehitys käsittää enimmäkseen ohjelmoinnin, jonka palvelin käsittelee. Vuonna 2009 julkaistu Node.js-alusta on mahdollista ajaa myös palvelimen puolelta ja antaa näin osaaville front end -kehittäjille mahdollisuuden käyttää jo hallitsemiaan taitoja back end -puolen hallinnassa. Tällaisia kehittäjiä, jotka hallitsevat niin front end - kuin back end -kehityksen, voidaan kutsua yleisemmin verkkokehittäjiksi. [3]

Front end -kehitys on kuitenkin hyvin nuori verkkokehittämisen konsepti. Vielä ennen CSS-kielen olemassaoloa sivustot rakennettiin HTML-taulukoiden avulla ja verkkosivujen suunnittelu ja niiden käytettävyys ei ollut lähellekään niin tärkeä kuin tänä päivänä. Verkkokehityskielen kehittyessä asiat ovat muuttuneet huomattavasti. Nykypäivän lukuisat eri näyttöpäätteet vaativat aivan oman osaajansa mukauttamaan verkkosivustot ja -sovellukset toimimaan jokaisella laitteella. Front end -tekniikat kehittyvät nopeasti niiden avoimuuden ja yhteisöllisyyden vuoksi. Avoimen lähdekoodin ratkaisuihin voi ottaa osaa kuka tahansa ja viedä kyseistä tuotetta eteenpäin oman osaamisen avulla. [4]

Työnkululla tarkoitetaan mallia, jonka työntekijä toistaa säännöllisesti työskennellessään. Front end -kehityksen työnkulkua parantamaan ja nopeuttamaan on tullut viime vuosina lukuisia ratkaisuja. Osa niistä automatisoi toimintoja ja työn vaiheita, osa taas laajentaa jo olemassa olevia työkaluja entistä monipuolisemmiksi. Niiden oikeaoppinen käyttäminen voi nopeuttaa projektin kulkua huomattavasti. Esimerkiksi Node.js:n päälle rakennettu tehtävän suorittaja (engl. task runner) Grunt on kehitetty juuri tätä varten. Tehtävän suorittajien tarkoituksena on nopeuttaa verkkokehittäjän työnkulkua vähentämällä vaivannäköä toiminnallisuuksien rakentamiseen. Se ryhmittää työt tehtäviin, jotka kootaan (engl. compile) automaattisesti samalla, kun työtä tehdään. Kehittäjän ei tarvitse itse asettaa tai ajaa toimintoja vaan tehtävän suorittaja hoitaa ne automaattisesti. [5,6]

3 TEKSTIEDITORIT

Yksi verkkokehittäjän tärkeimmistä työkaluista on tekstieditori. Tekstieditoreja on lukuisia erilaisia ja verkkokehittäjä voi hyödyntää lukuisilla eri tavoilla. Jotkin tekstieditorit ovat erikoistuneet jollain tietyllä kielellä kehittämiseen, toiset helpokäyttöisyyteen ja toiset taas muokattavuuteen. Tekstieditoreja löytyy ilmaisina, maksullisina ja avoimen lähdekoodin ratkaisuin. Kaikkia tekstieditoreja kuitenkin yhdistää se, että niillä kaikilla on tarkoitus luoda ja muokata tekstiedostoja, jotka pitävät sisällään muotoilematonta tekstiä (engl. plain text). Tekstieditoreilla ei ole tarkoitus muokata tekstin ulkonäköä kuten esimerkiksi Microsoft Wordilla. Tekstieditorien ei yleensä katsota olevan ohjelmointiympäristöjäkään (engl. IDE). Ohjelmointiympäristöllä tarkoitetaan ohjelmistoa, jolla kehittäjä voi rakentaa ohjelmia. Ohjelmointiympäristöt ovat usein ominaisuuksiltaan huomattavasti laajempia, suorituskyvyllisesti raskaampia ja usein myös monimutkaisempia käyttää kuin tekstieditorit. Ohjelmointiympäristöjä ovat esimerkiksi Microsoft Visual Studio sekä Java-pohjainen ohjelmointiohjelma Eclipse. Vaikka edellä mainituilla ohjelmilla pystyy myös kehittämään verkkosivuja, ovat tekstieditorit huomattavasti joustavampia, kevyempiä ja helpompia vaihtoehtoja verkkokehittäjälle. Tekstieditorien joustavuus tulee esille niiden tarjoamassa muokattavuudessa. Suosituimpia tekstieditoreja on helppo muokata juuri itselleen sopiviksi ja näin ollen tekstieditori pysyy ominaisuuksiltaan ja käyttöliittymältään selkeänä ja helpokäyttöisenä. [7]

Ilmaisista tekstieditoreista yleinen vaihtoehto Windows-alustoille on Notepad++. Kyseinen editori on kevyt ja helposti lisäosien kautta laajennettava, mutta saattaa olla käyttöliittymältään aloittelijoille sekava. Avoimen lähdekoodin tekstieditoreista suosittu vaihtoehto on Adoben tukema Brackets-editori. Brackets on tunnettu reaaliaikaisesta kehitysympäristöstään, jossa verkkokehittäjä voi nähdä koodinsa tuloksen reaaliajassa selaimessa samalla, kun työskentelee koodin parissa. Avoimuutensa takia se kehittyy jatkuvasti yhteisön tuella. Toinen mielenkiintoinen avoimen lähdekoodin tekstieditori-projekti on nimeltään Atom. Atomin kehityksen takana on GitHubin yhteisö. Kuka tahansa voi laajentaa

Atom -editorin toiminnallisuuksia ohjelmoimalla ne Node.js-alustan päälle käyttäen verkkokehityskieliä kuten CSS:ää ja JavaScriptiä. Myös kaupalliset suljetun lähdekoodin tekstieditorit ovat säilyttäneet suosionsa. Sublime Text -niminen tekstieditori on suljetun lähdekoodin omaava monipuolinen ja erittäin muokattava tekstieditori. Kukaan verkkokehittäjä voi muokata sen juuri itselleen sopivaksi. [8,9,10,11]

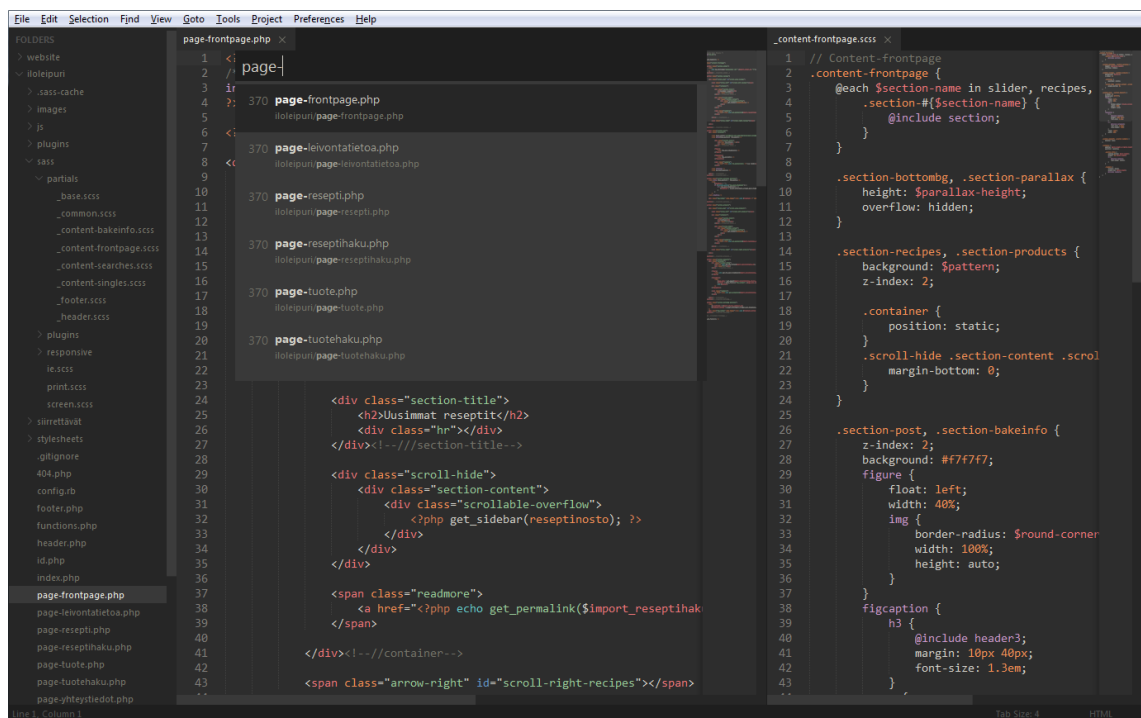
Sublime Text -editori on ainut maksullinen vaihtoehto koodin kirjoittamiselle tässä opinnäytetyössä vertailuun valituista tekstieditoreista. Se on myös ainut, josta löytyy versiot kaikille yleisimmille käyttöjärjestelmille. Atom on omaksunut omaan käyttöliittymäänsä Sublime Text -editorin komentopaletti toiminnallisuuden. Pythonilla kirjoitettu Sublime Text -editori kuitenkin käyttöliittymä toimii suavammin kuin Atom-editorin. Tämä johtuu siitä, että Atom on kirjoitettu verkko-ohjelmointikielien avulla ja niiden suorituskyky ei vastaa vielä esimerkiksi Pythonilla kirjoitettujen ohjelmien suorituskykyä. Brackets on myös rakennettu verkko-ohjelmointikielillä, mutta siinä on sisäänrakennettuna ominaisuus reaaliaikaiseen esikatseluun selaimessa. Brackets tukee Windows- ja OS X -käyttöjärjestelmiä. Linux tuki on kehitteillä. Notepad++ on saatavilla ainoastaan Windows-käyttöjärjestelmälle, mutta ilmaisena tekstieditorina se omaa rikkaan käyttöliittymän ja mahdollisuuden laajentaa sen ominaisuuksia lisäosilla. (Taulukko 1) [8,9,10,11,12]

Taulukko 1 Tekstieditorien vertailua

	Hinta	Avoin lähdekoodi	Kirjoitettu	Tuetut käyttöjärjestelmät	Esille nostettu ominaisuus	Suosittu lisäosa
Note- pad++	Ilmainen	Kyllä	C++	Windows	Välilehditetty käyttöliittymä	NppFTP
Brackets	Ilmainen	Kyllä	HTML, CSS, JavaScript	Windows, OS X	Reaaliaikainen esikatselu	Mark- down Preview
Atom (Beta)	Ilmainen	Kyllä	C++, Node.js	OS X	Verkkokehityskieliä rakennetut lisäosat	Color Picker
Sublime Text	\$70	Ei	Python, C++	Windows, OS X, Linux	Komentopaletti	Package Control

Sublime Text -editori vaikutti tutkimukseen otetuista editoreista sopivimmalta oppinäytetyön rakentamista varten. Kyseisen tekstieditorin ominaisuudet ja muokattavuus vakuutti kehitystiimin. Muokattavuuden lisäksi, Sublime Text -editorin suuri lisäosavarasto vaikutti suuresti kyseisen tekstieditorin valitsemiseen. Sublime Text -editorista oli myös tarjolla vakaa (engl. stable) 2. ohjelmaversio. Ohjelman vakaus on tärkeä valinta kriteeri. Tutkimusten pohjalta oppinäytetyön kehittämiseen valittiin siis Sublime Text -tekstieditori, joten sen ominaisuuksiin ja muokattavuuteen syvennyttiään tarkemmin.

Sublime Text on maksullinen tekstieditori saatavilla Windows-, OS X- ja Linux-käyttöjärjestelmille, joka on noussut kilpailemaan suosituimman tekstieditorin paikasta Notepad++:n kanssa. Sublime Text on helposti lähestyttävä ohjelma niin edistyneille kuin aloittelijoillekin. Sen selkeän käyttöliittymän, joustavuuden sekä monipuolisuuden myötä, Sublime Text on valikoitunut monien verkkokehittäjien suosikkitekstieditoriksi. (Kuva 1.) [13,14]



Kuva 1 Sublime Text -tekstieditorin käyttöliittymä

Verkkokehittäjä hyötyy monella tapaa Sublime Textin käytöstä. Kun kehittäjä ensimmäisen kerran avaa sivuston lähdekoodinsa Sublime Textillä, hän löytää

nopeasti editorin oikealta reunalta koodin pienoiskartan (engl. minimap). Se antaa yleisnäkymän koko koodista ja sen avulla verkkokehittäjän on helppo siirtyä haluamaan kohtaan omassa koodissaan. Suuremmilla näytöillä kehittäessä kannattaa ottaa käyttöön lisäkolumneja. Näin pystytään näkemään usean tiedoston sisältö samanaikaisesti. Tämä helpottaa erityisesti HTML:n ja CSS:n kehitystä, koska kehittäjä joutuu useasti vaihtamaan aktiivista tiedostoaan näiden kahden välillä. Vasemmalle reunalle verkkokehittäjä voi vielä avata koko projektikansionsa kaikkine alihakemistoineen. Sen avulla projektin kaikki tiedostot on helppo pitää hallinnassa. Kehittäjä voi myös helposti navigoida kaikkien projektitiedostojensa välillä avaamalla komentopaletin (engl. command palette) ja kirjoittamalla haluamansa tiedoston nimen editorin yläosaan ilmestyneeseen tekstikenttään. Tämän avulla usean eri projektiin liittyvien tiedostojen muokkaaminen sujuu nopeammin.

Sublime Textin käytön tehokkuus piileekin käyttöliittymän alla. Sublime Text on täynnä pikanäppäinyhdistelmiä, joita hallitsemalla verkkosivujen kehittäminen nopeutuu huomattavasti. Kaikki pikanäppäimet ovat myös täysin muokattavissa ja itse ohjelmoitavissa. Yksi hyödyllisimmistä pikanäppäin yhdistelmistä on komentopaletin lisäksi Ctrl + D/Cmd + D. Tämän avulla kehittäjä voi monistaa kohdistimensa ja valita monta samaa valittua tekstin osaa samanaikaisesti hallintaan. Tämä on kätevää silloin, kun halua muuttaa samanaikaisesti muuttujan nimen kaikissa kohdissa yhdessä tiedostossa. Myös esimerkiksi eri koodin kohtia voi laittaa kirjanmerkkeihin omalla pikanäppäinyhdistelmällään. Tämä tekee tiedoston sisäisestä navigoinnista vaivattomampaa. [15]

Aikaisemmin mainittu Sublime Textin joustavuus nousee uudelle tasolle siihen tarjolla olevien lisäosien ansiosta. Sublime Textin suosituin lisäosa on lisäosien hallintaan tarkoitettu Package Control. Sen avulla muiden lisäosien hallinta onnistuu sujuvasti Sublime Texin omalla komentopaletilla. Package Control -lisäosaa käyttäen kehittäjä voi helposti asentaa uusia tai poistaa vanhoja lisäosiaan. Toiseksi suosituin lisäosa on Emmet. Emmet nopeuttaa verkkokehittäjän projekteja tarjoamalla huomattavan määrän automatisoituja lisätoimintoja Sublime Text -editoriin. Kehittäjä voi esimerkiksi kirjoittaa

CSS-lyhenteen `m-10-20` tyyli-tiedostoonsa. Tämä lyhenne muuttuu toimivaksi tyylimääriykseksi `margin: -10px -20px`. CSS-lyhenteiden lisäksi Emmet automatisoi selainkohtaisten etuliitteiden (engl. vendor prefix) lisäämisen. Näiden lisäksi verkkokehittäjä voi hyötyä Emmetin käytöstä lukuisilla muilla tavoilla, kuten esimerkiksi täyttämällä kokonaisia HTML-tiedostoja vain yhdellä HTML-lyhenteitä sisältävällä rivillä tai luoda viisi riviä tyylimääriyksiä liukuväriille vain yhdellä lyhyellä komennolla. [16,17]

Emmetin lisäksi verkkokehittäjä voi laajentaa Sublime Textin toiminnallisuuksia lataamalla mitä tahansa muita lisäosia yli 2 000 lisäosan joukosta. Kehittäjä voi esimerkiksi ladata BracketHighlighter-lisäosan helpottamaan sulkeiden sijaintia koodissa tai LiveReloaderin, jolloin kehittäjä näkee tekemänsä muutokset välittömästi selaimessa. Toiminnallisuuksien lisäksi Sublimen ulkoasun ja koodin värimaailman (engl. color scheme) voi muokata itselleen sopivaksi. Theme - Soda on suosituin Package Control -lisäosan kautta ladattu teema. Oikean teeman ja värimaailman avulla verkkokehittäjä voi tehostaa työskentelyään, kun tekstieditori on myös visuaalisesti silmille sopiva. Tutkimukset ovat osoittaneet, että silmät väsyvät hitaammin, kun näytöltä lukee vaaleaa tekstiä tummalla pohjalla sävyjen kontrastin pysyen kuitenkin matalana. Nämä vaihtelevat ihmisestä riippuen ja jotkut kehittäjät valitsevat tekstieditorin pohjaväriksi mieluummin vaalean ja tekstin väriksi tumman. Tästä syystä verkkokehittäjän on syytä ottaa selvää, kumpi värimaailma sopii hänelle paremmin ja ladata vastaava värimaailma esimerkiksi Sublime Textiin jo ennen projektin alkua. [18]

4 KEHYKSET, KIRJASTOT SEKÄ CSS-ESIKÄSITTELIJÄT

Kehykset ja kirjastot sekoitetaan usein keskenään, mutta niillä on yksi selkeä ero: käyttäjä kutsuu kirjastoa, kun taas kehyksen on tarkoitus kutsua käyttäjää. Kehykset siis yleensä asettuvat koko verkkoprojektin lähdekoodin rungoksi, johon kehittäjä tekee tarvittavat muutoksensa. Kirjasto on vain kokoelma luokkia, joita käyttäjä kutsuu tarvittaessaan. [19]

Kun verkkokehittäjän tulee rakentaa sivulle esimerkiksi interaktiivisia toiminnallisuksia, voi kehittäjä lähteä rakentamaan niitä valmiiden kehyksien päälle. Monimutkaisissa verkkoapplikaatioissa suosittuna kehyksenä kehittäjät käyttävät AngularJS-kehystä. JavaScript-pohjaisten kehysten lisäksi CSS-pohjaisia kehyksiä on myös rakennettu helpottamaan verkkosivun kehitystä. CSS-pohjaiset kehykset tarjoavat tyylitiedostonsa yleensä myös jonkin CSS-esikäsittelijän muodossa. CSS-esikäsittelijät ovat taas kehitetty laajentamaan natiivin CSS:n ominaisuuksia ja ne mahdollistavat esimerkiksi muuttujien käytön tyylitiedostoissa. Kevyempien toiminnallisuksien rakentamisessa kehittäjä voi linkittää lähdekoodiin asiaankuuluvan kirjaston. Suosituin JavaScript-pohjainen kirjasto on jQuery. JQuery on myös näin ollen suosittu pohja JavaScript-pohjaisten lisäosien rakentamiselle. [20]

Kehittäjä voi näiden lisäksi ottaa käyttöön myös mallipohjan, joka on parhaita käytäntöjä (engl. best practise) huomioiden koodattu pohja. Yksi yleisimmin front end -kehittäjien käyttämistä mallipohjista on HTML5boilerplate. Sitä käyttämällä verkkokehittäjä saa projektin kuin projektin nopeasti aluilleen oikeaoppisine käytäntöineen. Mallipohjat myös helpottavat projektien eteenpäin vientiä ryhmätyöskentelyssä, koska sen ansiosta kaikilla on samanlainen selkeä lähtökohta ja esimerkiksi koodauskäytännöt ovat jo valmiiksi sovitut.

4.1 Kehykset

Kehyksiä on tarjolla lukuisia erilaisia niin back end - kuin front end -kehittäjille. Front end -kehittäjien suosituin CSS-pohjainen kehys on Twitter Bootstrap. Bootstrapin avulla verkkokehittäjä voi luoda monimutkaisinkin responsiivisen (engl. responsive) verkkosivun pienellä vaivalla. Responsiivinen verkkosivu muokautuu näytön kokoon niin, että sivuston käytettävyys säilyy. Kyseinen kehys pitää sisällään joukon valmiiksi määritettyjä luokan (engl. class) nimiä, joita oikein käyttämällä verkkosivuston työpöytä- ja mobiili-versio valmistuvat samanaikaisesti. Bootstrap tarjoaa myös joukon valmiita tyyliä lomakkeille, painikkeille, pudotusvetovalikoille (engl. dropdown menu), typografialle jne. Edistyneet kehittäjät yleensä haluavat asettaa tyylit kaikille elementeille itse, joten tätä varten Bootstrap-kehysten latauksen omaan käyttöön pystyy kokonaan muokkaamaan itselleen sopivaksi. Lataukseen pystyy esimerkiksi sisällyttämään ainoastaan responsiivisen ruudukon (engl. responsive grid). Tämä on yleinen tapa, kun halutaan responsiivinen sivusto omilla tyylitiedostoilla valmiiksi nopealla aikataululla. Vaihtoehtoinen front end -kehys Bootstrapille on Foundation. Foundation on responsiivinen kehys, jossa on valmis tuki CSS-esikäsittelijöille. Se on hyvä vaihtoehto, jos kehittäjä ei tarvitse joukkoa valmiiksi määriteltyjä käyttöliittymäelementtejä. Verkkokehittäjä, joka haluaa itse tyylittää kaikki elementtinsä, voi siis pärjätä paremmin Foundation-kehysten kanssa kuin Bootstrapin. Jos verkkokehittäjä ei ole luomassa responsiivista verkkosivua, voi se ladata 960 Grid System -kehysten nopeuttamaan sivuston kehitystä. 960 Grid System -kehysten ruudukko ei ole responsiivinen, mutta sen avulla onnistuu nopeasti esimerkiksi verkkosivujen tai applikaatioiden prototyyppien rakentaminen. [21,22,23]

Kun verkkokehittäjällä on edessään laajan verkkosivuston, -palvelun tai -applikaation rakentaminen, kannattaa hänen tutustua tarjolla oleviin JavaScript-pohjaisiin kehyksiin. Monipuolisen verkkoapplikaation rakentaminen ilman jäsenneltyä toimintamallia, voi johtaa sekavaan hakemistorakenteeseen ja epäloogisiin koodauskäytäntöihin. Tämä johtaa nopeasti myös siihen, että sivuston jatkokehitys on huomattavasti hankalampaa. Kun käyttöön ottaa JavaSc-

ript-pohjaisen kehyksen, koodin hallinta on huomattavasti helpompaa. Suosituimpia näistä on vuonna 2009 alkanut avoimen lähdekoodiin perustuva projekti AngularJS sekä seuraavana vuonna julkaistu kehys Backbone.js. [24,25,26]

4.2 Kirjastot

Kehittäjän suorittaessa monimutkaisempaa JavaScript-ohjelmointia kohtaa hän usein haasteita selainten eroavaisuuksissa. Jokin JavaScript-komento suoriutuu eri tavalla Googlen Chromella kuin Internet Explorerilla. Kirjastot ovat kehitetty helpottamaan tätä ongelmaa tarjoamalla valmiita funktioita kehittäjän käyttöön. Kirjastot ovat kehitetty niin, että ne toimivat selaimesta riippumatta. Suurilla ja monipuolisilla kirjastoilla, kuten jQueryllä, onnistuu toiminnallisuuksien tekeminen. Jos kehittäjä tietää tasan tarkkaan minkä ominaisuuden haluaa sivulle, voi hän ladata jonkin kevyemmän tiettyä toiminnallisuutta varten rakennetun JavaScript-kirjaston. Suosittuja JavaScript-kirjastoja on esimerkiksi fastclick.js. Kyseinen kirjasto eliminoi 300ms vasteajan, joka ilmenee mobiililaitella verkkosivuilla olevia linkkejä painaessa. Kaavioita varten on myös rakennettu useita JavaScript-pohjaisia kirjastoja. Yksi suosittu vaihtoehto on Chart.js. Se käsittelee syötettyä dataa JavaScriptillä ja tulostaa sen HTML5:n canvas-elementtiin. [27,28]

jQuery on suosituin JavaScript-pohjainen kirjasto. Sen avulla verkkokehittäjä voi manipuloida dokumenttioliomallia (engl. DOM), käsitellä tapahtumia (engl. event handling), animoida ja tehdä AJAX-kutsuja, jotka toimivat kaikilla selaimilla. Yli 60% verkon 100 000 suosituimmasta sivustosta käyttää jQueryä. jQueryllä DOM-elementtien valinta toimii samalla tavalla kuin CSS:ssä. Tämän takia jQuery on erittäin aloittelijaystävällinen. Verkkokehittäjä voi esimerkiksi valita DOM-elementin ja piilottaa sen hyvin lyhyellä koodin pätkällä `$(".class-name").hide()`. [29,30]

Kun verkkosivuprojektissa pitää esimerkiksi rakentaa kuvakaruselli (engl. image carousel), kannattaa ratkaisua lähteä jQueryn tarjoamista lisäosista. jQueryllä on myös tehty valtava määrä lisäosia, joilla myös aloittelevat verkkokehittäjät

voivat saada työstämästään sivustostaan huomattavasti interaktiivisemmat ja monipuolisemmat. Tämä tarkoittaa myös sitä, että edistyneidenkään kehittäjien ei kannata keksiä pyörää uudestaan. JQuerylle rakennetut lisäosat ovat parasta selata ensin läpi, ennen kuin alkaa itse rakentamaan haluttua toimintoa. Kuva-
karusellien lisäksi verkkokehittäjä voi hyödyntää esimerkiksi MixItUp nimistä jQueryä ja CSS3:n ominaisuuksia hyödyntävää lisäosaa luodessaan sekä visuaalisesti että toiminnallisesti näyttävää kategorisointi toiminnallisuutta verkkosivustolle. Tämän lisäosan avulla käyttäjä voi suodattaa näytettäviä elementtejä painamalla valitsemaansa kategorian nimeä. Elementit vaihtuvat ja uudelleen asettuvat sivulle animaation saattelemina. [31,32,33]

4.3 CSS-esikäsittelijät

Yleisimmät CSS-esikäsittelijät ovat Sass ja LESS. Näiden avulla verkkokehittäjä voi esimerkiksi käyttää sisäkkäisiä sääntöjä (engl. nested rules), muuttujia tai tallentaa kokonaisia tyylisääntöjä mixin-luokkiin. Näiden ominaisuudet mahdollistavat tyylitiedostojen helpomman hallinnan ja ylläpidon. Verkkokehittäjä voi esimerkiksi asettaa valitsemaansa CSS-esikäsittelijän mukaisessa tiedostossaan muuttujan (engl. variable) `blue` arvoksi värin hex-koodin `#012456`. Kun sivustolla käytetään haluttua sinisen sävyä elementeissä, voi kehittäjä käyttää hex-koodin sijasta muuttujaa. Kun jossain kohtaa kehitystä sinisensävyä joutuu muuttamaan hieman, kehittäjä voi vain muuttaa muuttujan arvoa, ja väri muuttuu kaikkiin elementteihin yhdellä kertaa. Muuttujien hyödyntäminen auttaa myös paljon elementtien ja typografiassa käytettävien fonttityylien kokoja määrittäessä. Oleellisena osana esikäsittelijöiden käyttöön kuuluu myös tyylitiedoston jakaminen osiin. Esimerkiksi useisiin Sass-aiheisiin parhaiden käytäntöjen -oppaisiin on kirjattu, että kehittäjän tulisi jakaa tyylitiedostonsa niin moneen osaan kuin on tarpeellista. Tämä tarkoittaa, että pohjalle, typografialle, ylätunnisteelle (engl. header), sisältöalueelle ja alatunnisteelle (engl. footer) luodaan omat tyylitiedostot. Tämän avulla tyylitiedostot eivät vain pysy paremmin hallinnassa, mutta ryhmässä työskentelystä tulee huomattavasti helpompaa. [34]

Sass ja LESS ovat toimintaperiaatteiltaan hyvin samanlaisia, mutta joitakin eroavaisuuksia niiden käytöissä kuitenkin on. Sass vaatii Ruby-ohjelmointikielen tuen kehitysalustalleen, kun taas LESS toimii JavaScriptin avulla. Molemmat CSS-esikäsittelijät tukevat sisäkkäisiä sääntöjä, mutta Sass vie tämän konseptin hieman pidemmälle salliessaan asettaa yksittäisiä ominaisuuksia sisäkkäin. Esimerkiksi eri reunoille (engl. border) voi määrittellä kullekin omat tyylinsä käyttäen sisäkkäisiä sääntöjä. Muuttujien kohdalla Sassin käyttäjät asettavat \$-merkin muuttujan eteen, kun taas LESS käyttää @-merkkiä muuttujan nimen edessä. Natiivi CSS kuitenkin käyttää jo @-merkkiä esimerkiksi media kysely -määritysten (engl. media query) edessä. Tästä syystä muuttujien käyttö Sass-käsittelijällä on loogisempaa ja helpommin ymmärrettävämpää. Verkkosivuston responsiivisuuden kehitys ja hallinta on myös huomattavasti helpompaa, kun käyttää CSS-esikäsittelijää. Sass ja LESS molemmat mahdollistavat media kyselyjen asettamisen sisäkkäin halutun CSS-valitsimen sisälle. Esimerkiksi Sass-kielellä media kysely voidaan asettaa halutulle luokalle seuraavalla tavalla.

```
.class-name {  
  
    width: 25%;  
  
    @media (max-width: 800px) {  
  
        width: 100%;  
  
    }  
  
}
```

Tällä tavoin HTML-elementin kaikki tyylilliset responsiiviset vaiheet voidaan määrittää samassa kohtaa ja tyylitiedoston hallinta helpottuu huomattavasti. [35]

Suoritettujen vertailujen pohjalta Sass-käsittelijä valittiin opinnäytetyössä käytettäväksi CSS-esikäsittelijäksi. Sen selkeys, monipuolisuus ja johdonmukaisuus nostivat sen paremmaksi valinnaksi näistä kahdesta. Selaimet eivät kuitenkaan pysty suoraan tunnistamaan .scss-päätteisiä Sass tiedostoja, vaan ne pitää ennen palvelimelle siirtoa koota CSS-tiedostoiksi. CSS-esikäsittelijöiden hallintaa varten on kehitelty paljon maksullisia ja ilmaisia sovelluksia ja kehyksiä. Ylei-

simpänä ilmaisen Sass-kehystenä käytetään Compass-kehystä. Se on komentorivillä käytettävä työkalu. Kätevimmillään Compass toimii silloin, kun se on asetettu automaattisesti kokoamaan kaikkia muokattuja tiedostoja. Tällöin verkkokehittäjän ei tarvitse murehtia koottujen CSS-tiedostojen ajantasaisuudesta. Koska tyyli on jaettu useaan eri tiedostoon, ryhmytyöskentelystä tulee huomattavasti helpompaa.

Compass-kehys pitää sisällään myös joukon valmiiksi määritettyjä mixin-luokkia, kuten esimerkiksi luokan nimeltään `background`. Käyttämällä `background`-luokkaa, verkkokehittäjän ei tarvitse huolehtia selainkohtaisten etuliitteiden lisäämisestä jokaisen taustan määrittämiselle. Compass-kehysten tarjoamaa `background`-luokkaa voidaan käyttää esimerkiksi seuraavalla koodipätkällä.

```
.class-name {  
  
    @include background(  
  
        image-url("foo.png"),  
  
        linear-gradient(top left, #333, #0c0),  
  
        radial-gradient(#c00, #fff 100px)  
  
    );  
  
}
```

Compass kokoo yllä olevan koodipätkän normaaliksi CSS-tiedostoksi, joka pitää sisällään erikseen kuusi riviä taustan määrittämiä. Jokainen taustan määrittäminen alkaa erillä selainkohtaisella etuliitteellä. Näiden valmiiden luokkien käyttäminen nopeuttaa verkkokehitystä huomattavasti. [33]

Myös Grunt-niminen tehtävän suorittaja pystyy rakentamaan CSS-esikäsittelijöillä tuotettua kieltä `.css`-tiedosto muotoon. Tämän lisäksi koodin minifiointi (engl. minify) onnistuu vain `config`-tiedostoja muokkaamalla. Maksullisista applikaatioista Codekit on suosittu. Codekit omaa käyttöliittymän, joten sen käyttö on helpompaa aloitteleville verkkokehittäjille. Codekit rakentaa CSS-esikäsittelijöiden lisäksi myös CoffeeScript-, Typescript-, Jade-, Haml-,

Slim-, Markdown- ja Javascript-kieliä samoin kuten tehtävän suorittaja Grunt.
[36]

5 SISÄLLÖNHALLINTAJÄRJESTELMÄ VERKKOSIVUN TAUSTALLA

Verkkosivuja rakennetaan yleisesti asiakkaille, joilla ei ole laajaa tietotekniikan tietämystä. Tällöin sivusto kannattaa rakentaa sisällönhallintajärjestelmän päälle. Sisällönhallintajärjestelmällä tarkoitetaan verkkotyökalua, jonka avulla sivuston ylläpitäjä pystyy hallinnoimaan sivujensa sisältöä, kuten tekstiä, lomakkeita, kuvia sekä muuta mediaa. Tämän avulla verkkokehittäjä voi määritellä sisällöllisesti päivitettäviä alueita ja sivuja. Näiden päivittämistä varten sisällönhallintajärjestelmillä on oma verkkokäyttöliittymä hallintapaneelille. Tätä käyttämällä sisältöä voi muokata vaivatta tekstinkäsittelyeditorin (engl. rich text editor) kanssa. Tämän lisäksi kuvien lataaminen sivulle ja niiden asettelu onnistuu vaivattomasti. Sisällönhallintajärjestelmä mahdollistaa sivuston päivittämisen ilman, että käyttäjä joutuu ollenkaan koskemaan sivun lähdekoodiin.

Kun kehittäjälle tulee eteen sisällönhallintajärjestelmän valinta, kannattaa ensin tutkia omaa teknistä taitotasoa. Eri sisällönhallintajärjestelmät vaativat kehittäjältään erilaisia verkkokehityksen taitotasoa. Suosituimpia sisällönhallintajärjestelmiä ovat Wordpress, Drupal ja Joomla. Drupal vaatii näistä eniten teknisiä taitoja verkkokehittäjältä, mutta antaa kehittäjälle parhaimmat mahdollisuudet kehittää monimutkaisia sivuja laajoilla luokitteluominaisuuksilla. Joomla on kehittäjälle helpommin lähestyttävä kuin Drupal, mutta vaativampi kuin Wordpress. Näistä Wordpress on kuitenkin tilastollisesti ylivoimaisesti suosituin. Jos on kyse on sivustosta, jossa asiakas päivittää sisällön itse projektin valmistuttua, on Wordpress hyvä vaihtoehto. Wordpress omaa selkeällä käyttöliittymällä varustetun hallintapaneelin, jonka kautta sivuston sisältöä pystyy helposti päivittämään. (Taulukko 2) [37,38]

Taulukko 2 Vertailua yleisimpien sisällönhallintajärjestelmien välillä [37]

	Drupal	Joomla	Wordpress
Tietoa	Drupal on tehokas ja kehittäjäystävällinen työkalu monimutkaisten verkkosivujen rakentamiselle.	Joomla on sekä kehittäjättä käyttäjäystävällinen vaihtoehto sisällönhallintajärjestelmäksi.	Wordpress aloitti blogialustana, mutta on kasvanut laajojenkin sivustojen sisällönhallintajärjestelmäksi.
Käytön helppous	Drupal vaatii näistä kolmesta kehittäjältään laajimmat tekniset taidot. Mahdollisuudet luoda kehittyneitä verkkosivuja on kuitenkin suurempi, kuin Joomla- tai Wordpress-järjestelmillä.	Joomla vaatii kehittäjältä vähemmän kuin Drupal, ja suhteellisen pienellä vaivalla kehittäjä pystyy luomaan monimutkaisiakin sivuja.	Nopean asennuksen jälkeen vähemmänkin teknistä taitoa omaava verkkokehittäjä voi luoda toimivat verkkosivut helposti.
Ominaisuudet	Drupal tunnetaan tehokkaimista luokitteluominaisuuksistaan.	Joomla on suunniteltu toimimaan yhteisöalustana sosiaalisen median ominaisuuksilla.	Verkkokehittäjät voivat helposti rakentaa verkkosivut asiakkaalle, joka myöhemmin pienellä opastuksella pystyy itse huolehtimaan sivuston sisällön päivityksestä.

Opinnäytetyössä käytettäväksi sisällönhallintajärjestelmäksi valittiin vertailujen pohjalta Wordpress. Valintaa tuki myös se, että Wordpress oli asiakkaalle ja kehittäjille jo entuudestaan tuttu. Avoimeen lähdekoodiin perustuva Wordpress on julkaisupainotteinen sisällönhallintajärjestelmä, joka on alunperin kehitetty blogien ylläpitoon. Viime vuosina Wordpress on kuitenkin kasvanut pelkästä blogialustasta käyttäjäystävälliseksi sisällönhallintajärjestelmäksi. Useat mainostoimistot ovat ottaneet Wordpressin mukaan yhdeksi käytettäväksi alustaksi rakennettavien sivustojen taustalle. Wordpressin on ottanut käyttöönsä jopa verkon suurimmat yritykset Google ja Facebook. [39]

Verkkokehittäjälle Wordpressin valitseminen alustaksi tarkoittaa teeman ja lisäosien rakentamista. Wordpressin teemalla tarkoitetaan kansiota, joka pitää si-

sällään kaikki tarvittavat tiedostot sivun ulkoasun muodostamiselle. Verkkokehittäjä voi luoda teeman lähes täysin käyttämällä vain front end -tekniikoita. Ainoastaan sisältöalueet määritetään sivupohjiin PHP-kutsuilla ja -silmukoilla. Kun sivustolle halutaan monimutkaisempia toiminnallisuuksia, kuten kalentereita tai lomakkeita, voi näitä etsiä yhteisön tekemistä lisäosista. Yli 30 000 lisäosan joukosta löytyy todennäköisesti haluttu toiminto, mutta tarvitsemansa lisäosan voi aina luoda myös itse. [40]

Wordpress on erityisen hyödyllinen front end -kehittäjille. Sen avulla kehittäjä voi suunnitella ja rakentaa verkkosivuston alusta loppuun itse. Back end -ohjelmointi ei ole vaadittavaa, koska Wordpressin asennus huolehtii valmiiksi kaikista tietokanta ja tietoturva-asioista. Kaiken tämän lisäksi Wordpress on ilmainen kaikille sen avoimen lähdekoodinsa ansiosta. Tästä syystä Wordpress on siis hyvä vaihtoehto esimerkiksi yksityisille freelancereille. Asiakasprojekteissa kustannukset voivat jäädä hyvinkin alas, koska monet front end -työkalut ovat Wordpressin tapaan ilmaisia. Asiakkaatkin ovat tyytyväisiä, kun he pystyvät sivuston valmistuttua hoitamaan itse kaikki sisällönpäivitykset.

6 SELAIMET JA VERKKOTYÖKALUT

Yksi tärkein verkkokehityksen osa-alue on testaus. Verkkokehittäjän tulee testata rakentamansa sivusto kaikilla yleisimmin käytetyillä selaimilla jo koko kehityskaaren ajan eikä vain projektin loppupuolella. Näin mahdolliset selainten väliset eroavaisuuden tulevat heti selville, eikä niitä aleta korjaamaan vasta päivä ennen sivuston julkaisua. Selainten käyttö osana verkkokehitystä on tärkeää projektin aikataulun kannalta. Moderni nykyajan selain on myös jo paljon muuten kuin vain verkkosivustojen selailua varten kehitetty sovellus. Selaimet kuten Mozilla Firefox, Chrome ja Internet Explorer 11, on kehitetty myös verkkokehittäjiä silmällä pitäen.

6.1 Kehittäjän työkalut selaimissa

Kaikkiin näihin edellä mainittuihin selaimiin on rakennettu kehittäjän työkalut. Niiden avulla kehittäjä pystyy tutkimaan (engl. inspect) verkkosivun elementtejä tähän tarkoitettuun työkalun avulla. Tämän tutkimustyökalun avulla pystytään valitsemaan haluttu verkkosivun elementti ja muokkaamaan esimerkiksi sen sisältöä tai tyylejä. Verkkotyökalut mahdollistavat reaaliaikaisen paikallisen kehityksen. Muutosten jälkeen muokattu tyylitiedosto pystytään lataamaan itselleen käyttöön. [41]

Kehittäjän työkalut pitävät myös sisällään paljon työkaluja, joilla voidaan tarkastella sivuston suorituskykyä ja mahdollisesti parantaa sitä näiden tarkastelujen pohjalta. Verkkonimisellä työkalulla voidaan tutkia jokaista sivuston verkkooperaatiota (engl. network operation), joka sisältää tietoa esimerkiksi ajoista, HTTP-pyyntöistä (engl. HTTP request), vastausten ylätunnisteista (engl. response headers) ja evästeistä (engl. cookie). Näitä tietoja monitoroimalla verkkokehittäjän on mahdollista tietää, mitä tietoa palvelimelta tulee, ja mitä sinne lähtee. Myös verkkosivuston nopeutta pystyy parantamaan tutkimalla latausnopeuksia ja vastaanotetun tiedon suuruuksia. Chromesta löytyvällä aikajana (engl. timeline) välilehdellä pystyy vielä tutkimaan, kuinka paljon muistia kyseinen verkkosi-

vu käyttää ja kuinka suurella ruudunpäivityksellä (engl. fps) sivustoa esitetään (engl. render). [42]

Kehittäjän työkaluista löytyy myös tehokas konsoli-työkalu (engl. console) JavaScript-komentosarjojen kehittämistä varten. Sen avulla virheiden paikantaminen ja poistaminen (engl. debug) on helppoa, ja se myös tukee omien komentosarjojen ajamisen suoraan verkkosivulla. Esimerkiksi `console.log()` komennolla voidaan suoraan tulostaa haluttuja muuttujia tai objekteja konsoliin. Chromessa voi vielä suodattaa konsolin sisältöä näyttämään ainoastaan esimerkiksi virheilmoitukset (engl. error message).

6.2 Verkkotyökalut

Verkko on täynnä resursseja, joista verkkokehittäjä voi etsiä apua omiin projekteihinsa. GitHub on isännöintipalvelu (engl. hosting service), joka tarjoaa ilmaiseksi avoimia varastoja rekisteröityneille jäsenille. Sen avulla on helppo jakaa omia projektejaan muiden käytettäväksi ja avustettavaksi (engl. contribute). Avoimen lähdekoodin ratkaisut ovat tämän vuoksi viime vuosina tulleet yhä suosittumaksi. Suosion johdosta niitä eniten isännöivän GitHubin käyttö on lisääntynyt huomattavasti. 23. joulukuuta 2014 GitHub ilmoitti rikkovansa 10 miljoonan varaston (engl. repository) määrän. Git-versionhallintaohjelman avulla kuka tahansa voi edistää avoimia projekteja kloonaamalla projektin omalle koneelle, tekemällä muutokset ja lähettämällä ne takaisin GitHubin isännöimään varastoon. Yhteisöllisyyden ansiosta avoimenlähdekoodin omaavat projektit kehittyvät nopeasti eteenpäin, huomattavasti nopeammin kuin suljetun lähdekoodin projektit. Githubista verkkokehittäjä voi ladata itselleen käytettäväksi esimerkiksi JavaScriptillä tai jQueryllä toteutettuja projekteja. Näitä kehittäjä voi sitten käyttää omissa verkkosivuprojekteissaan. [43]

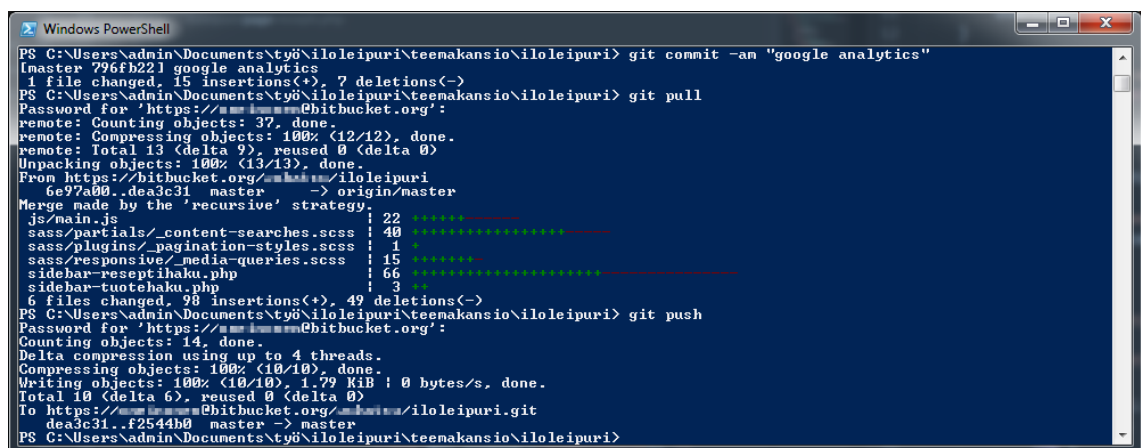
Internetissä on myös isännöitynä lukuisia erilaisia verkko JavaScript-ohjelmointiympäristöjä (engl. Online JavaScript IDE), jotka helpottavat front end -kehityskielen koodipätkien (engl. code snippet) kehittämistä sekä niiden jakamista. Nämä palvelut sisältävät yleensä editorit HTML:lle, CSS:lle ja

JavaScriptille. Näiden editorien avulla verkkokehittäjä voi esimerkiksi luoda prototyypin haluamastaan ominaisuudesta tai toiminnallisuudesta verkkosivuilleen. Kun koodinpätkä on valmis, voidaan sitä jakaa ja helposti pyytää mielipiteitä sen parantamiseen. Jotkin palvelut tarjoavat vielä mahdollisuuden varastoida useampia koodinpätkiä omalle tilille. Esimerkiksi Codepeniä käyttämällä tämä on mahdollista. Codepen mahdollistaa myös CSS:n kirjoittamisen kaikilla yleisimmillä CSS-esikäsittelijällä sekä JavaScriptin kirjoittamisen CoffeeScriptillä tai LiveScriptillä. Koodia kirjoittaessaan voi kehittäjä nähdä reaaliajassa tuloksensa omalla ruudullaan. Codepenin maksullinen Pro-versio tarjoaa vielä joukon muita editoria laajentavia ominaisuuksia. Pro-lisenssin omaavat käyttäjät näkevät reaaliaikaisesti tuloksensa myös kaikilla mobiililaitteilla samanaikaisesti. Tämän lisäksi Pro-käyttäjät pystyvät luomaan yksityisiä koodinpätkiä ja kehittämään niitä samanaikaisesti usean eri käyttäjän kanssa. Codepenin lisäksi ominaisuuksiltaan hieman eroavia samankaltaisia palveluja ovat esimerkiksi yksi ensimmäisistä verkko JavaScript-ohjelmointiympäristöistä nimeltään JSBin sekä suosittu JSFIDDLE. [44,45]

7 VERSIONHALLINNAN KÄYTTÖ OSANA VERKKOKEHITYSTÄ

Projektiryhmän kehittäessä samaa verkkosivustoa, voi se kohdata paljon ongelmia, kun muutoksia pitäisi tehdä samoihin tiedostoihin samanaikaisesti. Kaiken kaikkiaan tiedostojen jakaminen ja siirtely tiimin sisällä voi tuottaa huomattavia määriä ongelmia, jos tiedostoja siirretään verkkolevyn tai ftp-yhteydellä palvelimen kautta. Versionhallinnan käyttöönotto auttaa tässä asiassa. Versionhallinnalla tarkoitetaan järjestelmää, joka tallentaa tiedostoihin tehdyt muutokset niin, että näihin muutosversioihin voidaan palata myöhemmin. Tiedostojen varmuuskopiointi tapahtuu näin automaattisesti. Versionhallinnan avulla pystyy tarkastelemaan edellisiä muutoksia, kuka niitä on tehnyt ja milloin. Versionhallinta mahdollistaa aikaisempiin versioihin palauttamisen, jos esimerkiksi projekti-tiedosto turmeltunut (engl. corrupted). [46]

Versionhallinta on myös kevyttä, koska vain muuttuneet tiedostot lähetetään etävarastoon (engl. remote repository). Näin muutoksien tapahtuessa muiden ryhmän jäsenten ei tarvitse ladata koko projektia, he voivat ladata vain päivitettyt tiedostot paikalliseen varastoonsa (engl. local repository). (Kuva 2.)



```

Windows PowerShell
PS C:\Users\admin\Documents\työ\iloleipuri\teemakansio\iloleipuri> git commit -am "google analytics"
[master 796fb221 google analytics
1 file changed, 15 insertions(+), 7 deletions(-)
PS C:\Users\admin\Documents\työ\iloleipuri\teemakansio\iloleipuri> git pull
Password for 'https://[redacted]@github.com:':
remote: Counting objects: 37, done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 13 (delta 9), reused 0 (delta 0)
Unpacking objects: 100% (13/13), done.
From https://github.com/[redacted]/iloleipuri
6e97a00..dea3c31 master -> origin/master
Merge made by the 'recursive' strategy.
 js/main.js | 22 ++++++-----
 sass/partials/_content-searches.scss | 40 ++++++-----
 sass/plugins/_pagination-styles.scss | 1 +
 sass/responsive/_media-queries.scss | 15 ++++++
 sidebar-reseptihaku.php | 66 ++++++-----
 sidebar-tuotehaku.php | 3 ++
 6 files changed, 78 insertions(+), 49 deletions(-)
PS C:\Users\admin\Documents\työ\iloleipuri\teemakansio\iloleipuri> git push
Password for 'https://[redacted]@github.com:':
Counting objects: 14, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 1.79 KiB | 0 bytes/s, done.
Total 10 (delta 6), reused 0 (delta 0)
To https://github.com/[redacted]/iloleipuri.git
dea3c31..f2544b0 master -> master
PS C:\Users\admin\Documents\työ\iloleipuri\teemakansio\iloleipuri>

```

Kuva 2 Git-versionhallinnan käyttö Windows PowerShell -komentotulkilla

Yleisimmät versionhallinta ohjelmat ovat Linus Torvaldsin kehittämä Git, kolme vuosikymmentä olemassa ollut Subversion (SVN) ja Mercurial, joka on ilmainen

avoimen lähdekoodin ohjelmisto. Suositut verkkopalvelut kuten GitHub, GitLab ja BitBucket ovat kasvattaneet Git-versionhallinta ohjelman käyttöä uusien yritysten keskuudessa. Subversion on näistä ohjelmista vanhin, mutta sitä pidetään yleisesti hitaana ja siitä uupuu haaraumien (engl. branch) kehittyneempiä ominaisuuksia verrattuna kahteen muuhun vaihtoehtoon. Mercurial-ohjelman tavoitteena on säilyttää hyvä suorituskyky ja laajennettavuus versionhallinnassa. Mercurial on kuitenkin jäänyt hieman Gitin varjoon vaikka näiden kahden avoimen lähdekoodin ohjelmien julkaisuajankohdat olivat lähellä toisiaan. [47]

Git valittiin opinnäytetyön versionhallintaohjelmaksi, koska sen käyttäminen BitBucket-isännöintipalvelun kanssa on vaivatonta. Etävarastoksi oli helppo asettaa BitBucketissa luotu projekti. Gitin suosiolla sekä sen kattavalla dokumentaatiolla oli myös suuri vaikutus sen valintaan. Koska versionhallinnan käyttö oli front end -tiimille vielä melko uutta, selkeän dokumentaation merkitys oli suuri.

Verkkokehittäjä voi asettaa verkkokehitysprojektinsa Git-versionhallintaan antamalla kyseiselle hakemistolle komennon *git init* ja komennolla *git remote add origin remote_repository_url* määritetään sen etävaraston osoite. Nyt kehittäjä voi komennolla *git commit* asettaa paikalliset muutokset paikalliseen varastoon ja tämän jälkeen komennolla *git push* lähettää muutokset etävarastoon. Kun muutokset ovat etävarastossa, voi kuka tahansa projektiryhmän jäsen kloonata varaston omalle tietokoneelleen ja jatkaa projektin työstämistä. Koska vain tehdyt muutokset tiedostojen sisällä lähetetään etävarastoon muiden käytettäväksi, voi kaksi kehittäjää työskennellä samankin tiedoston parissa, kunhan nämä eivät muuta samoja koodin kohtia. Kun muutoksia tehdään samoihin kohtiin samanaikaisesti ja Gitin yhdistäminen (engl. merge) epäonnistuu, ilmoittaa Git ristiriidasta (engl. conflict). Ristiriita voidaan kuitenkin korjata helposti. Kyseisessä tiedostossa tulee valita manuaalisesti, kumpi muutos halutaan säilyttää. Ristiriidat voidaan välttää, kun jokainen kehitystiiminjäsen tietää mitä kukin tekee. [48]

8 ILOLEIPURI.FI-SIVUSTON RAKENTAMISEN ESIVALMISTELUT

Opinnäytetyön projektiosuuden toimeksianto saatiin Martinexilta. Martinex on vuodesta 1986 lähtien toiminut perheyrittys, joka toimii lahja- ja taloustavaroiden, tekstiilien sekä leikkikalujen valmistajana ja tukkuliikkeenä. Tavoitteena oli rakentaa reseptienhakusivusto Iloleipuri-leivontatuotesarjalle. Osa projektiryhmää keskittyi sivuston back end -puoleen ja osa front end -puoleen. Back end -tekniikat käsittävät projektissa SQL-, PHP- ja JavaScript-kielillä ohjelmoidut osuudet sivun taustalla sekä jQueryn mahdollistamia Ajax-kutsut. Front end -puolta rakennettiin HTML-, CSS- ja JavaScript-kielillä. Tämä opinnäytetyö keskittyy Iloleipuri-verkkosivuston front end -puolen ohjelmointiin. Samalla käsitellään tarkemmin, mitä front end -kehitystyökaluja ja -tekniikoita käytettiin ja kuinka niitä hyödynnettiin verkkosivustoa rakentamisessa. Front end -tiimissä oli myös toinen jäsen rakentamassa verkkosivustoa. Hän keskittyi enemmän verkkosivuston käytettävyyden sekä responsiivisuuden tutkimiseen ja kehittämiseen. [49]

Opinnäytetyön toteuttamista varten, Martinex Oy toimitti Iloleipuri-sivuston kehitettävän ulkoasun kuvatiedostoina tammikuussa 2014. Tiedostot sisälsivät ulkoasut jokaiselle päätason sivulle. Etusivulle toivottiin isoa kuvakarusellia, vaakasassa esitettävää luetteloa uusimpien reseptien sekä tuotteiden kuvista, kahta parallaksi-efektin (engl. parallax) omaavaa kuvaelementtiä sekä tekstile tilaa näiden välille. Resepti- sekä tuotehakusivujen tuli sisältää suodattimet hakujen rajaamiseksi. Hauista haluttiin myös dynaamiset, eli suodattimia käyttämällä tulokset rajautuisivat näiden mukaisesti reaaliajassa, eikä sivua siis tarvitsisi ladata aina uudestaan hakuheitojen muuttuessa. Info-sivulle haluttiin vielä yhteydenottolomake, jolla sivustolla kävijät pystyvät lähettämään sähköpostiviestejä koskien tuotetiedusteluja, viallisia tuotteita ja muuta palautetta. Tämän lisäksi koko sivusto haluttiin responsiiviseksi. Toivottu julkaisuajankohta määritettiin asiakkaan toiveesta maaliskuun 31. päiväksi, mutta sivuston responsiivisuus sovittiin toteutettavaksi vasta tämän jälkeen.

Ennen projektin aloittamista projektiryhmässä päätettiin yhteisistä toimintamalleista ja kehitystavoista. Projektinhallinnan viitekehyykseksi käyttöön otettiin scrum. Scrumin roolitukset suoritettiin projektiryhmän sisällä aikaisemman kokemuksen mukaisesti. Sprinttien pituus oli 2 viikkoa ja sprinttejä suunniteltiin olevan 5.

Tässä opinnäytetyössä tutkittiin ja vertailtiin erilaisia front end -kehitystyökaluja ja niiden pohjalta valittiin lloleipurin rakentamiseen käytettävät työkalut. Versionhallinnaksi käyttöön otettiin Git, ja varaston isännöintiin valittiin Bitbucket-verkkoisännöintipalvelun. Projektin front end -kehittäjien tekstieditoriksi valittiin käyttöön Sublime Text -editorin 2. version ja back end -puolella kehittäjät käyttivät Sublime Text -editorin lisäksi Notepad++-tekstieditoria. Sublime Text -editorin valintaan vaikutti sen muokattavuus ja laaja lisäosien valikoima. Notepad++:n valinneet tiiminjäsenet ottivat kyseisen tekstieditorin käyttöön, koska siitä oli näillä työntekijöillä eniten kokemusta aikaisempien projektien parissa. Sublime Text -editoriin asennettiin Package Control -lisäosan avulla BracketHighlighter-, Theme - Spacegray ja Sass-lisäosat. Natiivin CSS:n sijaan käytettiin CSS-esikäsittelijää Sassia ja tämän käyttöä helpottamaan valittiin Compass-kehys. Compassin sekä Gitin käyttöön valittiin Windows-käyttöjärjestelmän komentotulkki Windows PowerShell. Tiedonsiirto-ohjelmaksi käyttöön otettiin FileZilla ja sivuston testausta varten varmistettiin, että jokaiselta tietokoneelta löytyy yleisimmin käytetyt selaimet: Google Chrome, Mozilla Firefox, Internet Explorer sekä Safari. Tämän lisäksi varmistettiin mobiililaitteiden saatavuus testausta varten. Testaukseen valittiin iPhone5s-, iPad Air-, Macbook Air-, Samsung Galaxy S2- ja LG Nexus 4 -laitteet. Front end -tiimin sisällä sovittiin vielä yhteisistä ohjelmoinninkäytännöistä. CSS-luokkien ja Sassin muuttujien nimeämisperiaatteista sovittiin ensimmäisenä. Luokkien nimeäminen on tärkeää ja nimeämisessä tulee aina käyttää samaa kaavaa, esimerkiksi `list-item-category`. Näin CSS-valitsimena (engl. selector) voidaan käyttää vain yhtä luokan nimeä. Tämä mahdollistaa selaimelle nopeamman CSS-tiedoston ajamisen, joka taas tarkoittaa pienempiä sivun latausaikoja. [50]

9 ILOLEIPURI.FI-SIVUSTON TOTEUTUS

9.1 Työnkulku ja versionhallinnan käyttö projektiryhmässä

Opinnäytetyössä käytetty työnkulku koostui selaimessa käytettävien kehittäjän työkalujen käyttämisestä, koodin kirjoittamisesta tekstieditoriin, muokattujen tiedostojen siirrosta tiedonsiirto-ohjelmalla palvelimelle ja selaimella muokattujen elementtien tutkimisesta. Sass-tiedostojen rakentamista varten taustalla pyöri Compass-kehiksen komentotulkin komento *compass watch*. Tämän avulla kaikki muutetut *.scss*-tiedostot rakennettiin automaattisesti kokoonpanotiedostossa (engl. configuration file) määritettyyn sijaintiin. Työnkulkua helpottamaan olisi voitu käyttää myös Grunt-tehtävänsuorittajaa, mutta koska sen käyttö ei ollut kaikille projektiryhmän jäsenille tuttu, se päätettiin jättää pois käytettävistä front end -työkaluista.

Gitin käyttö versionhallintana helpotti ryhmätyöskentelyä huomattavasti. Kaksi eri front end -kehittäjää pystyi kehittämään sivustoa samanaikaisesti, jopa muokkaamaan samoja tiedostoja yhtä aikaan. Kun toisen tekemät muutokset ladattiin koneelle komennolla *git pull*, Git yhdisti automaattisesti kaikki muutokset paikalliseen varastoon. Muutosten jakaminen ryhmän sisällä olisi ollut huomattavasti hankalampaa, jos käytössä olisi ollut vain tiedonsiirto-ohjelma. Gitin käyttö oli kuitenkin kaikille front end -kehittäjille tuttu jo ennen projektin alkua, joten sen käyttöönotosta projektin versionhallinnaksi ei ollut eriäviä mielipiteitä.

9.2 Käytetyn CSS-esikäsittelijän hyödyntäminen

Sass-työkalun käyttö CSS-esikäsittelijänä on tehokasta, kun eri tyylialueen sisältävät määrytykset hajautettiin omiin *.scss*-tiedostoihinsa. Näin ryhmätyöskentelystä tuli vieläkin helpompaa, koska Git-versionhallinnassa törmättiin huomattavasti vähemmän ristiriitaihin sivuston tyyliä muokatessa. Tyylimäärytykset jaettiin kahdeksaan eri tiedostoon, jotka nimettiin seuraavasti:

- `_base.scss`
- `_common.scss`
- `_content-bakeinfo.scss`
- `_content-frontpage.scss`
- `_content-searches.scss`
- `_content-singles.scss`
- `_footer.scss`
- `_header.scss`.

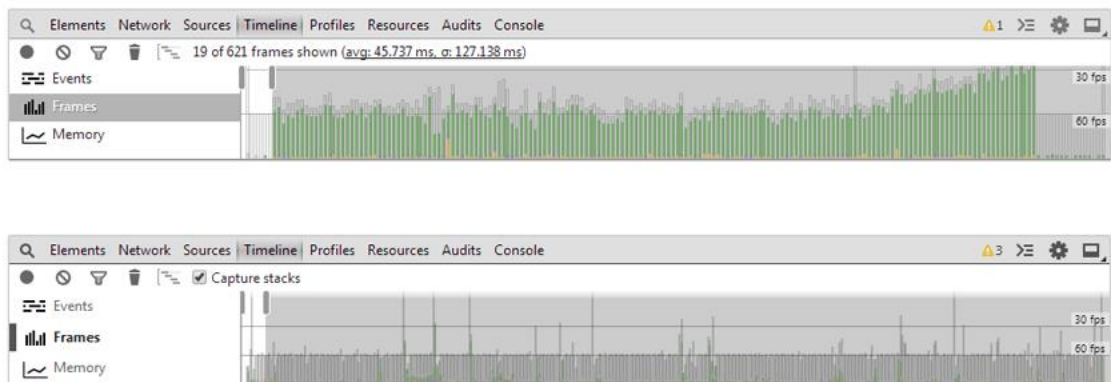
Screen.scss-tiedostossa on tuonti-lausekkeet (engl. import) muihin .scss-tiedostoihin. Ensimmäisenä tuodaan `_base.scss`-tiedosto, koska se pitää sisälleen kaikki muuttujat ja mixin-luokat, joita muut seuraavaksi tuotavat .scss-tiedostot käyttävät. Kun Compass rakentaa Sass-projektin, se yhdistää tähän screen.scss-tiedostoon kaikki muut tyylimäärittelytiedostot ja kokoaa kaiken yhteen screen.css-tiedostoon. Tämä mahdollistaa sen, että vaikka muutoksia on tehty useaan eri tyylitiedostoon, joutuu palvelimelle testausta varten siirtämään vain yhden rakennetun screen.css-tiedoston.

9.3 Kehittäjän työkalut selaimessa osana testausta

Tässä opinnäytetyössä kehitettävän verkkosivun jatkuvaa testausta suoritettiin kaikilla yleisimmillä selaimilla, mutta kehittäjän työkalut valittiin Google Chromen tarjonnasta. HTML-elementtien tutkimistyökalulla pystyi helposti testaamaan erilaisia tyylejä halutussa elementissä. Koska työkaluun kirjoitetut tyylimäärittelyt näkyivät reaaliaikaisesti selaimen dokumenttinäkymässä, tuli tästä nopeasti kehittäjän työkalujen käytetyin ominaisuus. Chromen konsoli (engl. console) toimi kätevimmin JavaScript-komentosarjoja kirjoittaessa. Sen avulla virheiden paikantaminen ja poistaminen helpottui huomattavasti. Etusivun vaakatasossa esitettävää resepti- sekä tuote-luetteloa varten rakennettiin jQuery-lisäosa. Tätä kehitettäessä konsoli oli erittäin tärkeä osa työnkulkua. Kyseistä lisäosaa rakennettaessa jouduttiin usein tarkistamaan esimerkiksi JavaScript-objektin sisältö. Konsoli pystyy näyttämään niin yksittäisten muuttujien arvoja kuin myös

kokonaisten objektien sisältöjä. Konsoliin tulostuvat myös selkeät virheilmoitukset, kun koodin syntaksi (engl. syntax) on virheellinen.

Kehittäjän työkaluilla pystyttiin helposti ylläpitämään sivuston hyvää suorituskykyä. Suorituskyvyllä tarkoitetaan sivuston sujuvaa käyttökokemusta ja korkeaa ruudunpäivityslukemaa. Chromen tarjoamalla kehittäjän työkalujen avulla huomattiin esimerkiksi CSS-tyylimäärittelyn `background-position: fixed` laskevan ruudunpäivitystä huomattavasti. Kun kyseinen määrittely poistettiin taustakuvista ja asetettiin käyttämään ladattua JavaScript-lisäosaa, suorituskyky parani huomattavasti. (Kuva 3.)



Kuva 3 Sivuston ruudunpäivityksen parantaminen kehitystekniikoita vaihtamalla

9.4 Valittujen kirjastojen ja JavaScript-pohjaisten lisäosien käyttöönotto

HTML-dokumentin otsakealueeseen (engl. head) linkitettiin Googlen isännöimän JavaScript-pohjaisen jQuery-kirjaston. Jotta sivuston latausajat parantuisivat, valittiin jQuery-versio, josta oli valmiiksi poistettu rivinvaihdokset ja sisennykset. Mahdollisten jQuery-pohjaisten lisäosien toimivuuden kannalta on oleellista linkittää se aina ensimmäisenä. JQueryyllä luotiin etusivun reseptien ja tuotteiden esittelyalueiden toimintaan oma scrollThis-niminen lisäosa. Sivuston muihin interaktiivisiin toimintoihin etsittiin niille tarkoitettuja JavaScriptillä rakennettuja lisäosia. Etusivun parallaksiefektien luomiseen käytettiin imageScroll.js-nimistä lisäosaa. Sen integroiminen sivulle onnistui helposti selkeän oppaan

ansiosta. Kyseinen lisäosa valittiin käytettäväksi sen suorituskykyisyyden takia. ImageScroll.js käytti kuvien liikuttamiseen CSS3:n transform ominaisuutta. Chromen tarjoamat kehittäjän työkalut mahdollistivat sivuston suorituskyvyn testaamisen. Kyseinen JavaScript-pohjainen lisäosa ei juuri laskenut sivuston ruudunpäivitystä, kun taas CSS:n absoluuttisen sijainnin avulla siirtelevät lisäosat laskivat sivuston suorituskykyä huomattavasti. Reseptisivun ja tuotesivun taittoon (engl. pagination) valittiin www.script-tutorials.com-sivuston laatiman oppaan pohjalta muokattu `imtech_pager.js` -lisäosa. Kyseisten sivujen hauelementin vierittämistä varten sivustolle linkitettiin vielä kaksi js-lisäosaa.

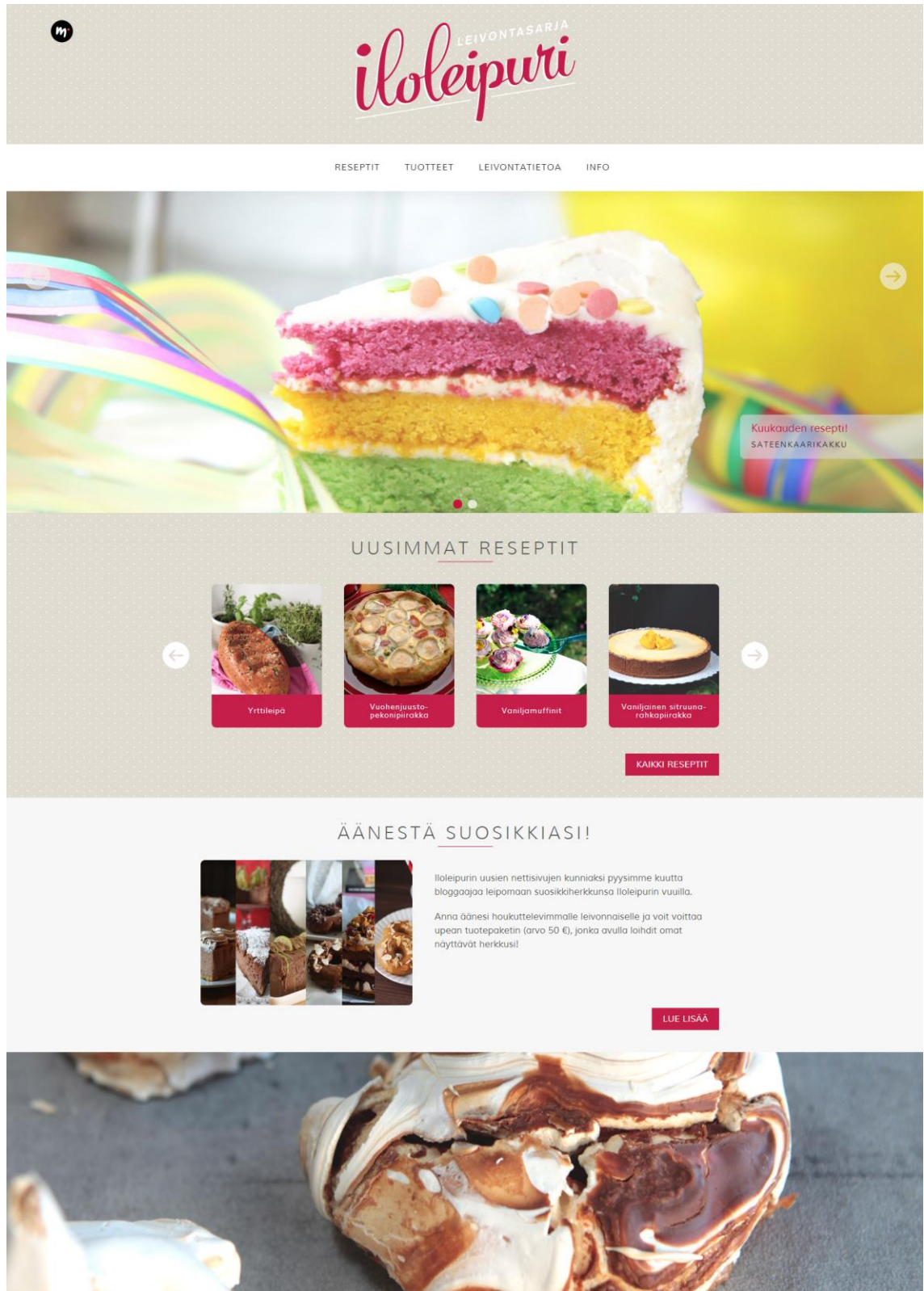
Google Fonts -sivuston tarjoamia ilmaisia fontteja käytettiin sivuston typografiassa. Google Fonts tarjoaa yli 600 avoimen lähdekoodin fonttiperhettä (engl. font family), jotka ovat kaikki optimoitu käytettäväksi verkossa. Iloleipuri-sivustolle valittiin asiakkaan kanssa käytettäväksi Lato- ja Muli-nimiset fontit ja niihin käytettävät fontin painot (engl. font weight). [51]

9.5 Wordpressin käyttö sivuston sisällönhallintajärjestelmänä

Yksittäisten sivujen sisällön ja median päivittämistä varten koko sivusto rakennettiin uutena teemana Wordpress-julkaisualustan päälle. Asiakkaan toiveiden mukaisesti reseptien ja tuotteiden päivittäminen tapahtui kuitenkin Wordpressin ulkopuolella Excel-taulukossa. Wordpressin avulla asiakas pystyy päivittämään helposti sivujen otsikkoja, sisältöjä ja lisäämään esimerkiksi uusia kilpailuja etusivulle. Etusivun kaikki kuvaelementit määritettiin niin, että asiakas pystyy itse tulevaisuudessa päivittämään ne. Koska Wordpress oli asiakkaalle jo entuudestaan tuttu, sen valinta muiden järjestelmien joukosta oli selvää alusta saakka. Info-sivun lomake toteutettiin Contact Form 7 -nimisen ilmaisen Wordpress-lisäosan avulla. Tämän lisäksi Info-sivun sisältö on täysin asiakkaan päivitettävissä.

9.6 Sivuston toteutus ja julkaisu

Opinnäytetyöhön käytettiin vajaa kolme kuukautta, joista itse toteuttamiseen noin kaksi kuukautta. Opinnäytetyön alun ensimmäiset viikot asiakas käytti ulkoasujen luomiseen. Lopullisen ulkoasun valikoiduttua kehitystiimi alkoi rakentaa sitä neljässä sprintissä. Scrumin aloitus- ja lopetuspalavereja pidettiin säännöllisesti sovittuina aikoina, ja jokaisen työpäivän alussa pyrittiin pitämään lyhyehkö palaveri tiimissä tehdyistä ja tulevista tehtävistä (engl. task). Kommunikointi oli kuitenkin helppoa koko projektin ajan, koska front end -kehitystiimi työskenteli suurimman osan ajasta samassa tilassa. Front end -työkalut, kuten Git ja projektin varaston isännöivä Bitbucket, mahdollistivat töiden tekemisen myös etänä. Sivuston testaamiselle varattu aika käyttämään yhä verkkosivun kehittämiseen. Tämä johtui suurilta osin sivuston sisällön puutteesta. Testaus lykkääntyi näin julkaisun yhteyteen, ja korjauksia jouduttiin tekemään jo julkaistuun sivuun vielä viikko julkaisun jälkeen. Työpöytäversion valmistuttua front end -tiimissä keskityttiin jatkokehittää sivuista responsiiviset. (Kuva 4.)



Kuva 4 Iloleipuri-sivuston etusivu

10 POHDINTA

Vaikka front end -kehitys on hyvin nuori kehityskonseptina on se ehtinyt kasvaa muutaman viime vuoden aikana hyvin paljon. Monelle kehittäjälle voi olla suuri työ pysyä tämän kehittyvän tekniikan perässä. Yritysten onkin kannattavinta valita hyvin tuetut ja omaan työympäristöön sopivat tekniikat käyttöön, joita voi käyttää vielä tulevien vuosien aikana. Yritysmaailmassa ei ole aikaa eikä resursseja tutkia ja testata jokainen uusi kehitys tai lisäosa ja kouluttaa niiden käyttöön työntekijöille. Verkkosivuja kehittävien yritysten on järkevintä valita pitkäikäisiä front end -työkaluja ja -tekniikoita käyttöönsä.

Freelancerina toimivien kehittäjien on helppo siirtyä testaamaan uusia kehitettyjä työkaluja, koska työskentelytiimissä on huomattavasti vähemmän ihmisiä. Tämä onkin suotavaa, sillä jos kehittäjä päätyy joskus isompaan yritykseen töihin, voi hän viedä mukaan tiedot ja taidot uusien tekniikoiden ja työkalujen käyttämisestä. Tämän opinnäytetyön verkkosivuprojektin kehittämistä varten otettiin käyttöön uutena työkaluna front end -kehitystiimin kaikille jäsenille Sass. Se oli yksi merkittävimmistä valinnoista tämän opinnäytetyön rakentamisen kannalta. Sen käytön nopea oppiminen ja hyödyllisyys verkkosivun kehityksessä vakuuttivat kehitystiimin jäsenet. Kyseisen CSS-esikäsittelijän käyttö sekä versionhallinnan vahvistunut osaaminen ovat taitoja, joita tiimin jäsenet tulevat varmasti käyttämään tulevissa töissään.

Kuten tämä opinnäytetyö on osoittanut, erilaisia työkaluja on rakennettu lähes jokaiselle front end -verkkokehityksen osa-alueelle. Tärkeintä ei kuitenkaan ole ottaa käyttöön kaikkia uusia työkaluja vaan oleellisinta on valikoida niistä itselleen sopivat. Sopivia työkaluja kannattaa etsiä omien teknisten taitojen rajoissa. Esimerkiksi CSS-esikäsittelijöihin tutustuminen kannattaa jättää myöhemmäksi, jos CSS-kieli ei ole vielä kokonaan hallussa. Kun muutaman verkkosivustoprojektin jälkeen HTML-elementtien tyylittäminen onnistuu vaivattomasti CSS-kielellä, voi seuraavassa projektissa kokeilla Sass-käsittelijää. Compass-kehityksen vuoksi Sass on LESS-käsittelijää aloittelijaystävällisempi. Avoimen-

lähdekoodin kirjastot löytyvät tänä päivänä helposti GitHubista. GitHubin suosion vuoksi Git toimii myös kätevimmin versionhallintaohjelmana.

Ohjelmistoja ei kannata lähteä vaihtamaan, jos kokee olevansa tehokkaimmillaan käyttäessään niitä, joiden käyttöön on tottunut. Esimerkiksi tämän opinnäytetyön projektia toteutettaessa osalla kehittäjistä oli tekstieditorina käytössä Notepad++ ja osalla Sublime Text. Testaukseen sekä virheiden paikantamiseen ja poistamiseen osa jäsenistä käytti Mozilla Firefoxin kehittäjien työkaluja toisten suosiossa Chromen vastaavia ominaisuuksia. Tämä verkkosivuston kehitysprojekti kuitenkin osoitti, ettei yksilön ohjelmistojen valinnan tarvitse olla sama muiden jäsenten kanssa. HTML-tiedostojen kehittäminen sekä Sassin käyttäminen CSS-esikäsittelijänä onnistui kaikilta jäseniltä tekstieditorista riippumatta. Tärkeintä on sopia yhteisistä ohjelmointikäytännöistä ja käytetyistä tekniikoista. On täysin kiinni kehittäjän tottumuksista ja kokeilunhaluisuudesta, millä näitä käytäntöjä ja tekniikoita käyttää.

LÄHTEET

- [1] Nielsen, J. & Norman, D. 2014. The Definition of User Experience. Viitattu 1.4.2014 <http://www.nngroup.com/articles/definition-user-experience/>
- [2] Codesido, I. 2009. What is front-end development? Viitattu 1.4.2014 <http://www.theguardian.com/help/insideguardian/2009/sep/28/blogpost>
- [3] Node.js 2014. Node.js Viitattu 2.4.2014 <http://nodejs.org/>
- [4] Milo, J. 2014. A Front-End Workflow For The Evolving Web. Viitattu 2.4.2014 <http://www.portent.com/blog/design-dev/a-front-end-workflow-for-the-evolving-web.htm>
- [5] Business Process Management (BPM) Center of Excellence (CoE). 2009. Glossary.
- [6] Gunsolo, M. 2013. Get Up And Running With Grunt. Viitattu 2.4.2014 <http://coding.smashingmagazine.com/2013/10/29/get-up-running-grunt/>
- [7] Pettit, N. 2013. Which Text Editor Should I Use? Viitattu 5.4.2014 <http://blog.teamtreehouse.com/which-text-editor-should-i-use>
- [8] adobe/brackets 2014. adobe/brackets. Viitattu 31.5.2014 <https://github.com/adobe/brackets>
- [9] Atom. 2014. A hackable text editor for the 21st Century. Viitattu 5.4.2014 <https://atom.io/>
- [10] Brackets. 2014. Code the Web. Viitattu 5.4.2014 <http://brackets.io/>
- [11] Boudreaux, R. 2011. Open source Notepad++ is a solid contender among text editors. Viitattu 5.4.2014 <http://www.techrepublic.com/blog/web-designer/open-source-notepad-plus-plus-is-a-solid-contender-among-text-editors/>
- [12] Backslash. 2012. 11 Plugins to improve Notepad++ Viitattu 4.6.2014 <http://www.backslash.gr/content/blog/misc/11-notepad-plusplus-plugins>
- [13] Sublime Text. 2014. Sublime Text. Viitattu 27.4.2014 <http://www.sublimetext.com/>
- [14] Henry, A. 2014. Most Popular Text Editor: Notepad++. Viitattu 27.4.2014 <http://lifehacker.com/five-best-text-editors-1564907215/1567287033/+alanhenny>
- [15] Kinder, K. 2013. Sublime Text: One Editor to Rule Them All? Viitattu 7.4.2014 <http://www.linuxjournal.com/content/sublime-text-one-editor-rule-them-all>
- [16] Emmet. 2014. Emmet — the essential toolkit for web-developers. Viitattu 7.4.2014 <http://docs.emmet.io/>
- [17] Package Control. 2014. Package Control. Viitattu 7.4.2014 <https://sublime.wbond.net/>
- [18] Anthony. 2011. When to Use White Text on a Dark Background. Viitattu 10.4.2014 <http://uxmovement.com/content/when-to-use-white-text-on-a-dark-background/>
- [19] Wang, X. 2014. Library vs. Framework? Viitattu 11.4.2014 <http://www.programcreek.com/2011/09/what-is-the-difference-between-a-java-library-and-a-framework/>
- [20] Similiar Tech. 2014. Top JavaScript Technologies. Viitattu 15.4.2014 <http://www.similartech.com/categories/javascript>

- [21] 960 Grid System. 2014. 960 Grid System. Viitattu 31.5.2014 <http://960.gs/>
- [22] Foundation. 2014. Foundation. Viitattu 31.5.2014 <http://foundation.zurb.com/>
- [23] Gerchev, I. 2013. Top 10 Front-End Development Frameworks. Viitattu 10.4.2014 <http://www.sitepoint.com/top-10-front-end-development-frameworks/>
- [24] Hess, S. 2013. ANGULARJS DIRECTIVE DESIGN MADE EASY. Viitattu 27.4.2014 <http://seanhess.github.io/2013/10/14/angularjs-directive-design.html>
- [25] Backbonejs. 2014. Introduction. Viitattu 11.4.2014 <http://backbonejs.org/#introduction>
- [26] Orsini, L. 2014. Angular, Ember, And Backbone: Which JavaScript Framework Is Right For You? Viitattu 12.4.2014 <http://readwrite.com/2014/02/06/angular-backbone-ember-best-javascript-framework-for-you#awesm=-oAKcUe1hO2jSPt>
- [27] Chart.js. 2014. Chart.js. Viitattu 13.4.2014 <http://www.chartjs.org/>
- [28] Fastclick. 2014. Polyfill to remove click delays on browsers with touch Uis. Viitattu 14.4.2014 <https://github.com/ftlabs/fastclick>
- [29] Chalkley, A. 2014. Why jQuery is the Most Popular JavaScript Library. Viitattu 15.4.2014 <http://blog.teamtreehouse.com/jquery-popular-javascript-library>
- [30] jQuery. 2014. What is jQuery? Viitattu 16.4.2014. <http://jquery.com/>
- [31] Mixitup. 2014. Introducing MixItUp 2. Viitattu 17.4.2014 <http://mixitup.io/>
- [32] jQuery. 2014. The jQuery Plugin Registry. Viitattu 17.4.2014 <https://plugins.jquery.com/>
- [33] Hampton-Smith, S. 2014. The 21 best jQuery plugins. Viitattu 19.4.2014 <http://www.creativebloq.com/jquery/top-jquery-plugins-6133175>
- [34] Sass-lang. 2014. Sass Basics. Viitattu 19.4.2014 <http://sass-lang.com/guide>
- [35] Coyier, C. 2012. Sass vs. LESS. Viitattu 31.5.2014 <http://css-tricks.com/sass-vs-less/>
- [36] Bradley, S. 2012. Sass And LESS: An Introduction To CSS Preprocessors. <http://www.vanseodesign.com/css/css-preprocessors/>
- [37] Rackspace Support. 2013. CMS Comparison: Drupal, Joomla and Wordpress. Viitattu 22.5.2014 http://www.rackspace.com/knowledge_center/article/cms-comparison-drupal-joomla-and-wordpress
- [38] Similiar Tech. 2014. Top CMS Technologies. Viitattu 27.4.2014 <http://www.similartech.com/categories/cms>
- [39] Wordpress. 2014. About WordPress. Viitattu 19.4.2014 <https://wordpress.org/about/>
- [40] Wordpress. 2014. Plugin Directory. Viitattu 19.4.2014 <https://wordpress.org/plugins/>
- [41] Secrets of the Browser Developer Tools. 2014. Secrets of the Browser Developer Tools. Viitattu 1.6.2014 <http://devtoolsecrets.com/>
- [42] Google. 2014. Evaluating network performance. Viitattu 20.4.2014 <https://developers.google.com/chrome-developer-tools/docs/network>
- [43] Briandoll. 2014. 10 Million Repositories. Viitattu 20.4.2014 <https://github.com/blog/1724-10-million-repositories>

- [44] Sharp, R. 2014. JS Bin for Collaborative JavaScript Debugging. Viitattu 20.4.2014 <http://remysharp.com/2008/10/06/js-bin-for-collaborative-javascript-debugging/>
- [45] Codepen. 2014. PRO Features. Viitattu 20.4.2014 <http://codepen.io/pro>
- [46] Chacon, S. 2009. Pro Git.
- [47] Daityari, S. 2014. Version Control Software in 2014: What are Your Options? Viitattu 22.5.2014 <http://www.sitepoint.com/version-control-software-2014-what-options/>
- [48] The Odin Project. 2014. Git 101. Viitattu 22.4.2014 <http://www.theodinproject.com/web-development-101/git-basics>
- [49] Martinex. 2014. Yritys. Viitattu 19.5.2014 <http://www.martinex.fi/yritys/>
- [50] MDN. 2014. Selectors. Viitattu 22.4.2014 https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_started/Selectors
- [51] Google. 2014. Google Fonts. Viitattu 23.4.2014 <https://www.google.com/fonts>