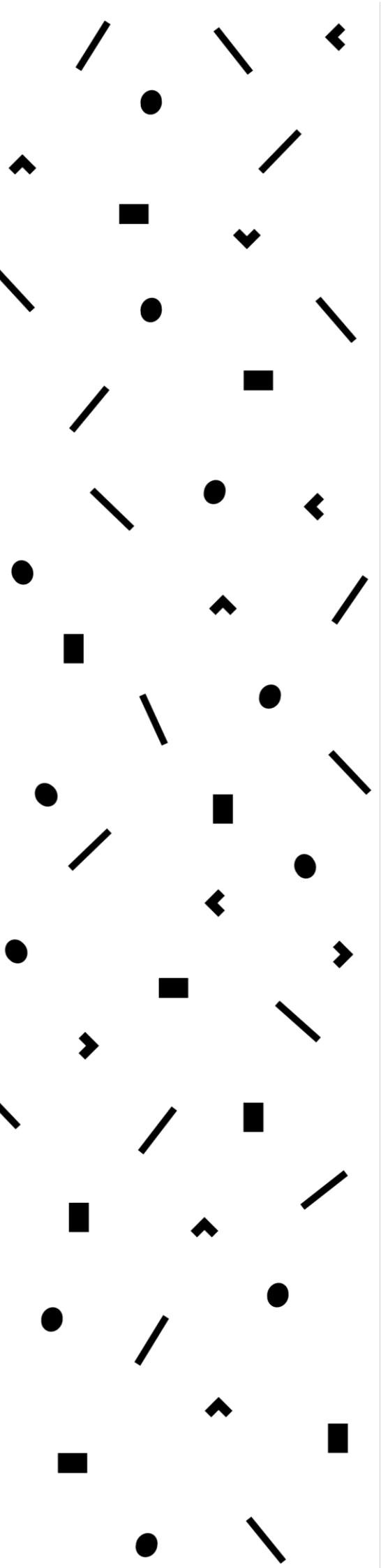


LAMK Lahden ammattikorkeakoulu
Lahti University of Applied Sciences

SIIRTYMINEN RESPONSIIVISEEN VERKKOSIVU-SUUNNITTELUUN

LAHDEN
AMMATTIKORKEAKOULU
Tekniikan ala
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka
Opinnäytetyö
Kevät 2015
Mika Blomberg

Lahden ammattikorkeakoulu
Tietotekniikan koulutusohjelma

BLOMBERG, MIKA:

Siirtyminen responsiiviseen verkkosivusuunnitteluun

Ohjelmistotekniikan opinnäytetyö, 41 sivua

Kevät 2015

TIIVISTELMÄ

Responsiivisella verkkosivusuunnittelulla tarkoitetaan tapaa suunnitella ja työstää verkkosivu niin, että sivu mukautuu näyttölaitteelle sopivaksi tämän koosta riippumatta. Aikaisemmin responsiivisten verkkosivujen sijaan tehtiin erillisiä web-mobiilisivuja, joiden ylläpitämiseen meni paljon aikaa. Responsiivista verkkosuunnittelua varten on kehitetty monia erilaisia frameworkkeja, joista muutamia käsitellään tässä opinnäytetyössä.

Opinnäytetyön tarkoituksena oli käydä läpi verkkosivujen teon perusteita, selvittää mitä responsiivinen verkkosuunnittelu on ja miten eri responsiiviset frameworkit toimivat. Työn käytännön osuutena työstettiin verkkosivut viidellä eri tekniikalla, jotta frameworkien välinen vertailu oli mahdollista. Toimeksiantajana opinnäytetyössä toimi DevNet Oy

Työssä käsitellään aluksi muutamia verkkosivutekniikoita, joiden ymmärtäminen on oleellista verkkosivujen suunnittelussa. Tämän jälkeen käydään läpi verkkosivujen suunnittelua ja sen vaiheita. Suunnittelun jälkeen käydään läpi responsiivista verkkosuunnittelua ja tämän pahimpia kilpailijoita. Lisäksi tarkastellaan kolmea eri responsiivista frameworkia, joita käytetään demosivuston tekemisessä. Työn lopussa suoritetaan frameworkien vertailu ja valitaan paras mahdollinen framework DevNet Oy:tä varten, jolla voitaisiin korvata nykyinen 960 Grid System-framework.

Sopivimmaksi frameworkiksi valittiin Bootstrap, sillä se sisältää kaiken sen, mitä responsiiviselta frameworkilta vaaditaan eikä erillisiä css-tiedostoja tarvitse käydä läpi, jotta web-ohjelmoija löytää oikeat tyylit. Bootstrap noudattaa myös mobile first-periaattetta, mikä helpottaa nettisivujen ulkoasun suunnittelua. Mobile first-ajattelu myös ennaltaehkäisee monia mahdollisia yhteensopivuusongelmia, joiden korjaamiseen menisi runsaasti aikaa.

Asiasanat: responsiivinen verkkosuunnittelu, Unsemantic, Pure, Bootstrap

Lahti University of Applied Sciences
Degree Programme in Information Technology

BLOMBERG, MIKA: Transitioning to Responsive Web Design

Bachelor's Thesis in software engineering, 41 pages

Spring 2015

ABSTRACT

Responsive Web Design is a way to design and make websites that adapt to the display screen despite its size. Before responsive websites there was a separate website for mobile phones which took a lot of time to maintain. There have been developed several types of frameworks for responsive web design, some of which are dealt with in this thesis.

The purpose of this thesis was to go through some basics of web development and to clarify what responsive web design is and how different responsive frameworks work. As a practical part of the thesis, a website was made using five different techniques so that a comparison between different frameworks could be done. The company that commissioned the thesis was DevNet Oy.

A few website techniques are presented at the beginning of the thesis because it is important to understand different techniques in website designing. After that website designing and its phases are dealt with as well as responsive web design and its worst competitors. The demo website was made using three different responsive frameworks. The most suitable framework for DevNet Oy was chosen. That framework could replace the currently used 960 Grid System -framework.

The most suitable framework was Bootstrap, which contains everything that a responsive framework is required to have and there is no need to go through separate CSS files to find the right styles. Bootstrap is also based on the principle "mobile first", which makes website layout design much easier. The mobile first idea prevents many compatibility problems where fixing would take a lot of time.

Key words: Responsive Web Design, Unsemantic, Pure, Bootstrap

SISÄLLYS

LYHENTEET JA SANASTO

960 Grid System	Rakennekehys web-sivujen asetteluun
Body	Sisältöalue, johon voidaan sijoittaa esimerkiksi tekstiä ja kuvia
CMS	Content Management System, sisällönhallinta-järjestelmä
CSS	Cascading Style Sheets, tyyliohjekieli, jota käytetään verkkosivujen tyylittelyyn
DOM	Document Object Model, suomeksi dokumenttioliomalli
Footer	Nettisivun alatunniste
Framework	Sovelluskehys
Grid	Taittopohja tai ruudukko, johon responsiivinen nettisivu rakennetaan
Header	Nettisivun ylätunniste
HTML	Hypertext Markup Language, internetin kuvauskieli, jolla voidaan kuvata hypertekstiä
JavaScript	Oliopohjainen ohjelmointikieli selainskriptien tekemiseen
Mobile first	Ajatus, jonka mukaan verkkosivujen tekeminen aloitetaan mobiililaitteesta ja laajennetaan tarvittaessa suurempiin näyttökokoihin
Navigation bar	Valikko, jonka avulla siirrytään sivulta toiselle
Sidebar	Sivuvalikko

1 JOHDANTO

Ihmisten internetin käyttö eri laitteilla on lisääntynyt valtavasti. Internetiä voidaan käyttää nykyään tietokoneen lisäksi myös älypuhelimella, tabletilla ja jopa televisiolla. On tärkeää, että tarvittavia nettisivuja voidaan käyttää millä laitteella tahansa ilman, että käyttökokemus kärsii.

DevNet Oy on IT-alan yritys, joka tarjoaa esimerkiksi verkkosivu-, mainostoimisto- ja ulkoistuspalveluita. Yrityksen tarjoamia tuotteita ovat muun muassa DevPro 3 -toiminnanohjausjärjestelmä ja eCMS-julkaisujärjestelmä. DevNetillä on kaksi toimipistettä, jotka sijaitsevat Lahdessa ja Jyväskylässä. (DevNet Oy 2015a.)

Erityisesti mobiililaitteella verkkosivua käytettäessä perinteisellä tavalla tehtyä nettisivua voi joutua suurentamaan ja vierittämään sivua runsaasti. Syynä ongelmaan on, että sivustot ovat alun perin suunniteltu käytettäväksi ja toimivaksi vain tietokoneen näytöllä. Kaikille alustoille optimoituja nettisivuja voidaan tuottaa responsiivisen verkkosuunnittelun avulla.

Responsiivisen verkkosuunnittelun ansiosta verkkosivuille ei tarvitse suunnitella erillistä web-mobiilisivua vaan jo olemassa oleva sivu pystyy mukautumaan oikeanlaiseksi ja kokoiseksi näytön koon mukaan. Responsiivinen verkkosivu on myös hyvä vaihtoehto natiivi-sovellukselle, sillä se on alustariippumaton ja sen käyttökokemus ja ulkoasu voidaan tehdä hyvin samanlaiseksi kuin natiivi-sovelluksen. (Leiniö 2012.)

Opinnäytetyön tarkoituksena on selvittää, mitä responsiivinen verkkosuunnittelu on ja millä tavalla se vaikuttaa verkkosivujen suunnitteluun käytettävään aikaan. Opinnäytetyössä vertaillaan myös kolmea responsiivista frameworkia ja pohditaan, mikä niistä olisi paras vaihtoehto korvaamaan käytössä olevan 960 Grid System -frameworkin.

Työn tueksi toteutetaan verkkosivut frameworkien kokeilemista varten. Verkkopalvelussa on sivuja, jotka on tehty frameworkien lisäksi myös perinteisellä tavalla sekä käyttämällä 960 Grid System -ruudukkoa.

Verkkosivuja on tarkoituksena käyttää apuna frameworkien vertailussa esimerkiksi käytännöllisyyden ja toiminnallisuuden osalta.

2 VERKKOSIVUTEKNIIKAT

Verkkosivujen tekemisen peruspilareihin kuuluvat kolme yleisintä web-tekniikkaa: HTML, CSS ja JavaScript. Kyseisten web-tekniikoiden perusteet tulee ymmärtää, jotta perinteisten sekä responsiivisten verkkosivujen tekeminen on mahdollista.

2.1 HTML

HTML on standardoitu merkintäkieli, jota käytetään verkkosivujen luomiseen. HTML toimii verkkosivujen runkona ja määrittelee sivuston sisällön sekä rakenteen ja ulkoasun käyttämällä erilaisia merkintöjä ja attribuutteja (W3C 2015). Tim Berners-Lee kehitti HTML:n vuonna 1989, jolloin kielen tarkoituksena oli pääasiassa kuvata www-sivuston rakennetta eikä niinkään sen ulkoasua mutta myöhemmin sivujen kirjoittajat halusivat vaikuttaa yhä enemmän dokumenttiensa ulkoasuun. (Raggert 1998.)

HTML:n viimeisimmät versionumerot ovat 4.01 ja 5. Tärkeimmät uudistukset HTML5:ssä olivat DOM-puurakenne määrittely sekä audio-, video- ja canvas-elementit. (W3C 2014.)

2.2 CSS

CSS on tyyliohjekieli, jota käytetään web-dokumenttien, kuten HTML- ja XML-tiedostojen, tyylittelyyn. CSS:n avulla voidaan muokata verkkosivun graafista ulkoasua asettamalla eri attribuuteille tyyliehdotuksia, kuten fonttikoko, tekstinväri ja rivinväli. (Saarikumpu 2014.)

World Wide Web Consortium (W3C) julkaisi CSS:n vuonna 1996. Alussa monet selaimet eivät tukeneet CSS:ää tai jotain sen osaa, mikä teki sivustojen suunnittelusta haastavaa. Kun selaimet vihdoin alkoivat tukea CSS:ää, ohjelmoijat ja suunnittelijat eivät olleet halukkaita käyttämään sitä sen epäluotettavuuden vuoksi. Ongelmista huolimatta CSS on nykyään oleellinen osa web-ohjelmointia. (CSS Neuse 2015.)

2.3 JavaScript

JavaScript on oliopohjainen ohjelmointikieli, jota käytetään dynaamisten toimintojen lisäämiseen verkkosivuille. JavaScript on kaikista käytetyin selainskriptien tekemiseen käytetty kieli, sillä useimmat internet-selaimet tukevat sitä. JavaScriptillä voi esimerkiksi ennakkotarkistaa lomakkeelle syötetyn datan ja hakea palvelimelta päivämäärän ja kellonajan (Korpela 2009).

JavaScriptin kehitti Brendan Eich vuonna 1995, ja se otettiin käyttöön osana Netscape-selainta vuonna 1996. Java oli ohjelmointikielenä erittäin suosittua 1990-luvulla, ja tämän vuoksi JavaScript-nimi viittaa siihen. JavaScript ei kuitenkaan nimestään huolimatta ole Javaa. (W3C 2012.)

3 VERKKOSIVUJEN SUUNNITTELU

3.1 Verkkosivujen sisältö ja kohderyhmä

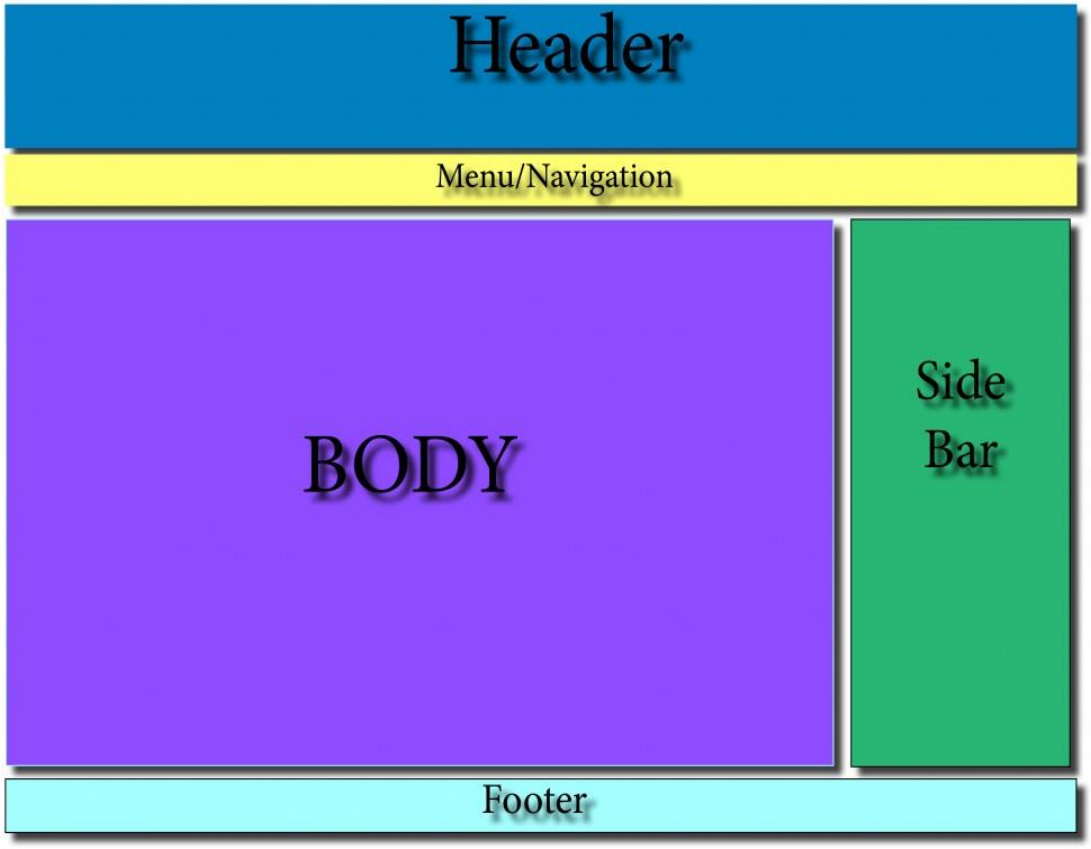
Verkkosivujen suunnittelun alussa on tärkeää huomioida, miksi ja kenelle verkkosivut tehdään. Verkkosivujen ulkoasu ja sisältö ovat erilaisia riippuen siitä, onko kyseessä yksityishenkilön, yrityksen vai yhdistyksen nettisivut. Yksityishenkilön sivusto voi olla blogimainen ja hyvinkin vapaamuotoisesti kirjoitettu, kun taas yrityksen tai yhdistyksen verkkosivun sisältö on hyvin asiallisesti ja virallisesti kirjoitettua. (Verkkotaikurit 2015).

Yritysten verkkosivut voivat sisältää esimerkiksi ajankohtaisia ilmoituksia, hinnaston sekä yhteystiedot. Yritysten sivujen tulee olla sisällöltään erittäin laajat, sillä niiden pitää kattaa nykyisten jo asiakkaana olevien ihmisten sekä potentiaalisten uusien asiakkaiden tarpeet. Verkkosivujen tulee kuitenkin välttää liian selvän mainosmaisen olemuksen luomista, sillä se saattaa karkottaa potentiaaliset uudet asiakkaat sivuilta. Yritysten verkkosivut tulee myös olla ajantasalla, sillä päivittämättömät sivut luovat kuvan huonon ja väliinpitämättömän kuvan yrityksestä. Yritysten sivut ovat niin sanottuja kohdennettuja mainoksia asiakkaille, jotta heidän olisi helppo ostaa yrityksen palveluita. Tämän vuoksi onkin tärkeää tehdä yritykselle verkkosivut, sillä ne ovat usein esimmäinen kontakti asiakkaan ja yrityksen välillä. (Verkkotaikurit 2015.)

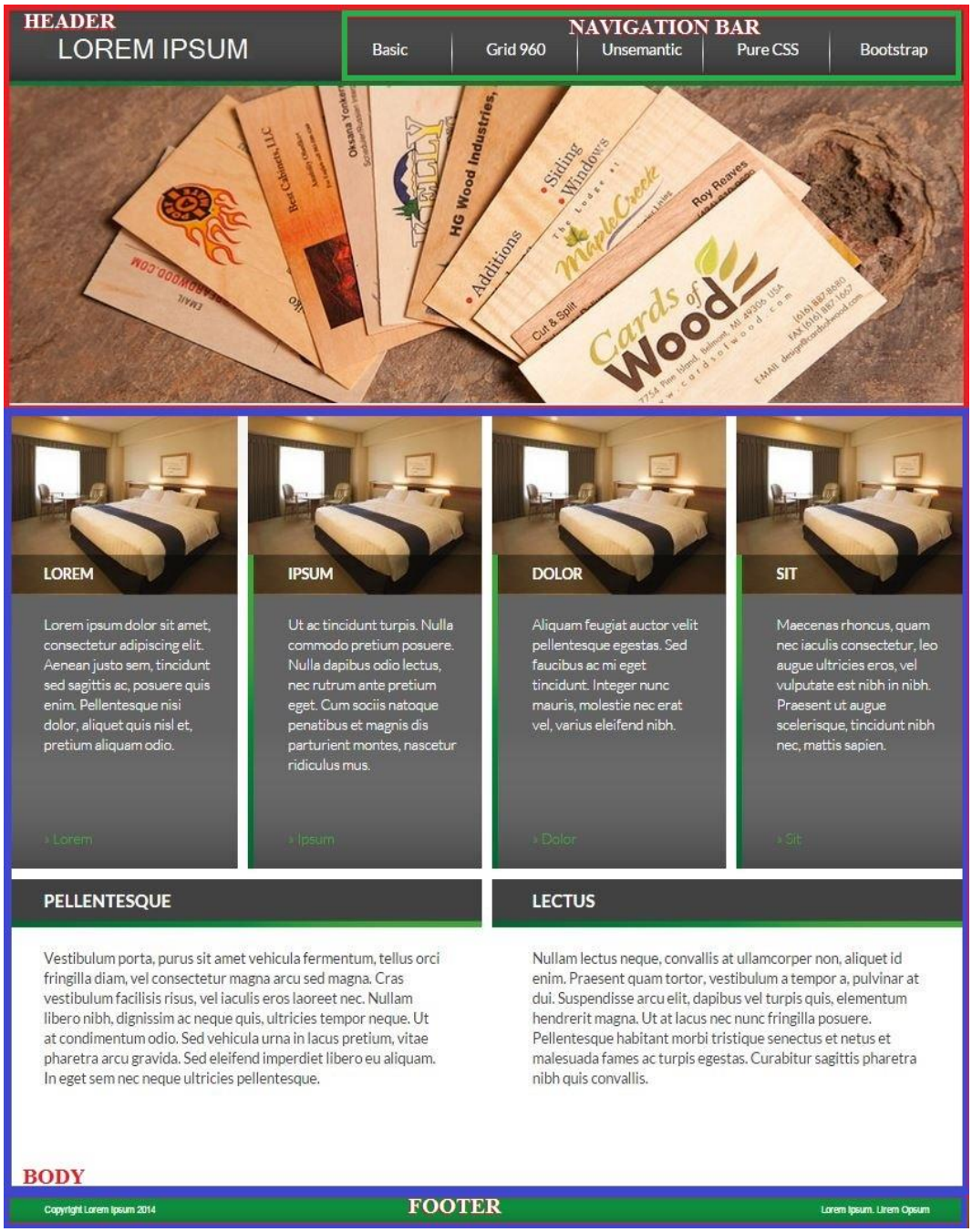
Yksityishenkilöiden verkkosivut voivat olla esimerkiksi henkilökohtainen blogisivusto tai fanisivusto, jonka kohteena on jokin tv-sarja tai bändi. Tämänkaltaisen sivujen ensisijainen ominaisuus on sisältö, sillä on tärkeää keksiä monipuolista ja omaperäistä kirjoitettavaa, jotta ihmiset kiinnostuvat nettisivuista. Esimerkiksi fanisivuilla käydään pääosin mieleenkiintoisten kuvien ja tekstien eikä niinkään ulkoasun vuoksi. Kiinnostavaa sisältöä voi olla esimerkiksi ravintola-, kirja- ja elokuva-arvostelut tai erilaiset käyttöoppaat ja ohjeet. (Verkkotaikurit 2015.)

3.2 Verkkosivujen rakenne

Vaikka maailmassa on runsaasti erinäköisiä verkkosivuja, niiden pääasiallinen rakenne on kuitenkin sama (kuvio 1). Verkkosivun ylä- ja ja alalaidassa ovat ylä- ja alatunniste eli header ja footer. Headerin alapuolella on yleensä valikko (navigation bar), jonka avulla voi helposti siirtyä sivulta toiselle. Navigation barin ja footerin välissä oleva alue on nimeltään body, johon laitetaan nettisivun sisältö. Bodyn molemmille puolille on myös mahdollista laittaa sivuvalikko (sidebar), johon voidaan sijoittaa esimerkiksi toinen valikko (QR8 Design Blog 2014). Sivujen perusrakenne on kuvattuna kuviossa 2.



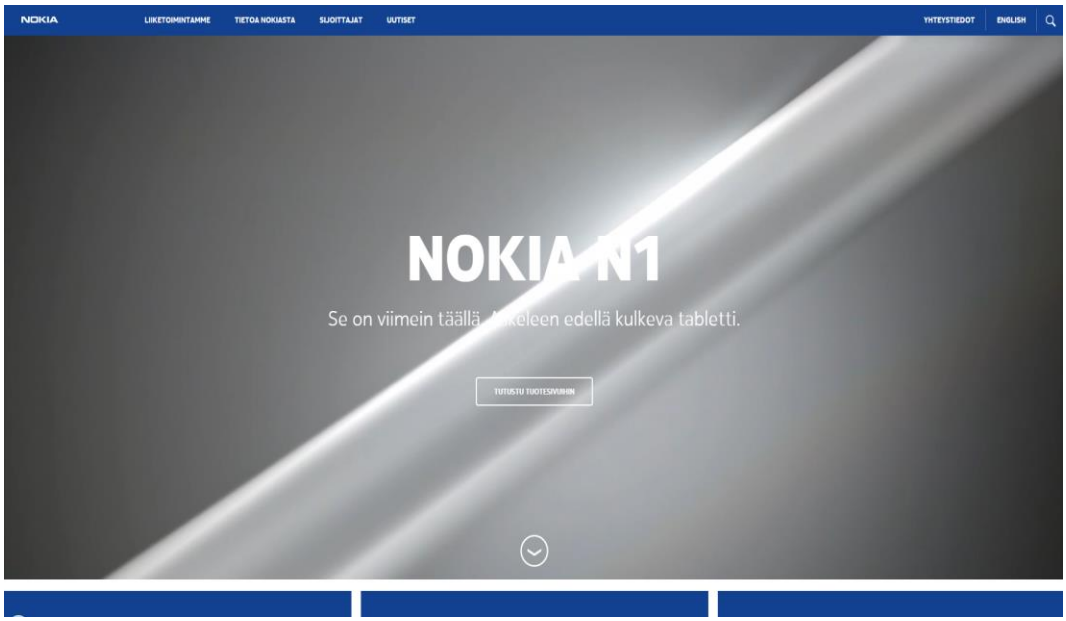
KUVIO 1. Nettisivujen perusrakenne (QR8 Web Design Blog 2014)



KUVIO 2. Demosivun rakenne

3.3 Verkkosivujen ulkoasu

Verkkosivujen ulkoasu on tärkeä osa verkkopalvelua, sillä kävijä kiinnittää siihen ensimmäisenä huomionsa. Ulkoasun tulee olla selkeä, mielenkiintoinen sekä mieleenpainuva. Hyvin suunniteltu sivusto mukailee yrityksen imagolle tärkeitä teemoja, fontteja ja värejä (kuvio 3). (Heinonen 2015.)



KUVIO 3. Nokian verkkosivun ulkoasu (Nokia 2015)

Verkkosivujen ulkoasun suunnittelussa tulee muistaa, että eri selaimet näyttävät verkkosivun eri tavalla. Tämän vuoksi sivusto tulisi suunnitella niin, ettei se näytä kovinkaan erilaiselle eri selaimilla. Suunnittelussa tulee myös huomioida kuvien, skriptien ja animaatioiden käyttö, sillä mitä enemmän niitä on sivuilla, sitä pidempään sivuston lataaminen kestää. Tämä on erityisesti rasitteena pienitehoisilla päätelaitteilla ja hitailla internet-yhteyksilä. (Virtanen 2015.)

Verkkosivujen ulkoasun kannalta tärkeä asia on myös esteettömyys. Esimerkiksi pelkästään Yhdysvalloissa on näköhäiriöisiä ihmisiä noin 7,3 miljoonaa ja Suomessakin vastaava luku on noin 80 tuhatta. Esteettömyyden jotkin muodot ovat erittäin hankalia, ja ne vaativat monimutkaisia palveluita, kuten kuvaselotuksen. Esteettömyys voidaan jakaa psyykkisiin ja fyysisiin rajoitteisiin sekä ohjelmallisiin ja laitteellisiin rajoitteisiin. (Ekonoja, Lahtonen & Mäntylä 2011.)

Käyttäjän rajoitteet tulee ottaa huomioon verkkosivujen suunnittelussa. Tietokone tarjoaa itsessään monia helppokäyttötoimintoja, kuten suurennuslasin ja näyttönäppäimistön. Suunnittelija voi myös itse vaikuttaa esteellisyyden ehkäisyyn käyttämällä suuria fontteja, selkeitä painikkeita

sekä mahdollisimman vähän häiritseviä väriyhdistelmiä. (Ekonoja ym. 2011.)

Erityisesti yrityksille on kannattavaa huolehtia sivuston esteettömyydestä, sillä se mahdollistaa suuremman kävijämäärän verkkosivuilla ja tällöin myös potentilaalisten asiakkaiden määrä kasvaa. Esteettömyydestä huolehditaan yleensä nykyaikaisten standardien avulla, mikä helpottaa sivuston ylläpitoa ja laajennettavuutta. Lisäksi verkkosivujen hakukonetulokset voivat parantua, mikä onkin erityisen tärkeää, jotta ihmiset löytävät yrityksen. (Ekonoja ym. 2011.)

3.4 Hakukoneoptimointi

Hakukoneoptimointi tarkoittaa prosessia, jossa tarkoituksena on päästä parhaaseen mahdolliseen verkkopalvelun hakukonenäkyvyyteen. Pääosin hakukoneoptimoinnilla tarkoitetaan Googlen hakutuloksia, minkä vuoksi sitä kutsutaan myös Google-optimoinniksi. (Ala-Harja 2014b.)

Käytännössä hakukoneoptimoinnilla tarkoitetaan sitä, että sivu, jolla toteutetaan optimointia, sijoittuu Googlen etusivulle 10 ensimmäisen hakutuloksen joukkoon haluttua hakusanaa käyttämällä. Optimointiprosessissa tutkitaan, mitkä ovat yritykselle keskeisiä ja tuottavia hakusanoja. Verkkosivua muokataan ja optimoidaan niin kauan, kunnes sivusto saadaan 10 parhaan hakutuloksen joukkoon ja panos-tuotto-suhde on kannattavaa. (Ala-Harja 2014b.)

Hakukoneoptimoinnissa on neljä peruselementtiä:

1. rakenne
2. sisältö
3. domain
4. linkitys.

Rakenteeseen littyviä asioita ovat esimerkiksi otsikoinnit sekä meta-tiedot. Title eli otsikkotagi on tärkeä osa hakukoneoptimointia, sillä Google-

haussa se näkyy hakutuloksen otsikkona. Otsikko-tagiin kannattaa keksiä lause, joka sisältää tärkeän avainsanan, jotta ihmiset avaavat juuri kyseisen sivun muiden joukosta. On myös tärkeää, että verkkosivulla on ainoastaan yksi pääotsikko (h1-elementti). Elementtiin kannattaa laittaa heti ensimmäiseksi yritykselle tärkein avainsana. Meta-tietoihin kirjoitetaan kuvaus (description) sekä avainsanat (keywords) (kuvio 4). Kuvauksen pituus on maksimissaan 156 merkkiä pitkä, ja siinä tulee olla tiivistettynä sivuston oleellinen sisältö. Avainsanojen merkitys on pienentynyt runsaasti, sillä niitä on käytetty jo pitkään väärin. Keskeisiä sanoja kannattaa silti laittaa sivulle 3 - 5 kappaletta. (Ala-Harja 2014a.)

```
<head>
    <title>Kimmon vaihtoautot Oy</title>
    <meta name="description" content="Kimmon vaihtoautot myy autoja Lahdessa">
    <meta name="keywords" content="vaihtoautot, henkilöautot, pakettiautot">
    <meta name="author" content="kimmonvaihtoautot.fi">
</head>
<body>
    <h1>Vaihtoautot Lahdesta edullisesti!</h1>
</body>
```

KUVIO 4. Esimerkki meta-tagien käytöstä kotisivuoptimoinnissa

Verkkosivun sisällön tulee olla sujuvasti ja ytimekkäästi kirjoitettua sekä tarkoituksenmukaista. Esimerkiksi autokaupan sivuilla asiakasta todennäköisesti kiinnostaa enemmän vaihtoautot kuin autokaupan historia. Lisäksi tarpeellinen ja erottuva sisältö auttaa nousemaan hakukonetuloksissa, sillä mitä useammin tietyllä hakusanalla päädytään samalle sivulle, sitä korkeammalle hakutulos nousee Googlen hakukoneessa. (Ala-Harja 2014a.)

Domain-nimi kannattaa suunnitella niin, että se sisältää jonkin tärkeän avainsanan. Lisäksi Google arvostaa enemmän vanhoja kuin uusia Domaineja, ja tämän vuoksi Domain-nimi kannattaakin hankkia mahdollisimman varhaisessa vaiheessa (Ala-Harja 2014a.)

Sivustolle on myös tärkeää, että se on linkitetty mahdollisimman moneen sivuun. Google nostaa hakutuloksissa korkeammalle sellaisen sivun, joka

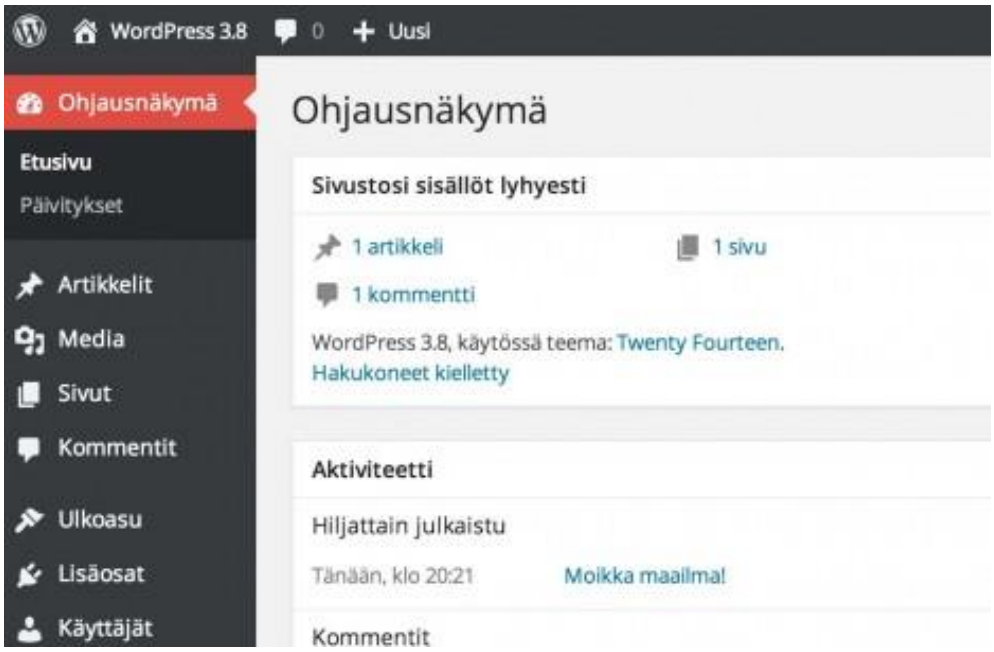
on linkitetty tietyllä avainsanalla monella eri verkkosivulla (Ala-Harja 2014). Esimerkki linkittämispalveluihin erikoistuneista yrityksistä on Fonecta (Fonecta 2015).

3.5 Julkaisujärjestelmä

Julkaisujärjestelmä eli CMS (Content management system) on verkkopalveluun asennettava järjestelmä, jonka tarkoituksena on yksinkertaistaa sivuston sisällön muokkaamista. CMS:n käyttäjän ei tarvitse välttämättä osata ohjelmoida, sillä sisällön lisääminen julkaisujärjestelmällä mukailee hyvin paljon tekstieditorin käyttöä. Joitain CMS-järjestelmiä voidaan käyttää sisällön muokkaamisen lisäksi myös uusien toimintojen kuten sivuvalikon lisäämiseen. (Ubinet 2015.)

Markkinoilla on runsaasti erilaisia julkaisujärjestelmiä, joista osa käyttää suljettua lähdekoodia ja osa käyttää avointa lähdekoodia. Avoimen lähdekoodin julkaisujärjestelmä on lisenssimaksuton, ja sitä voi jatkokehittää sekä hyödyntää kuka tahansa. (Ubinet 2015.)

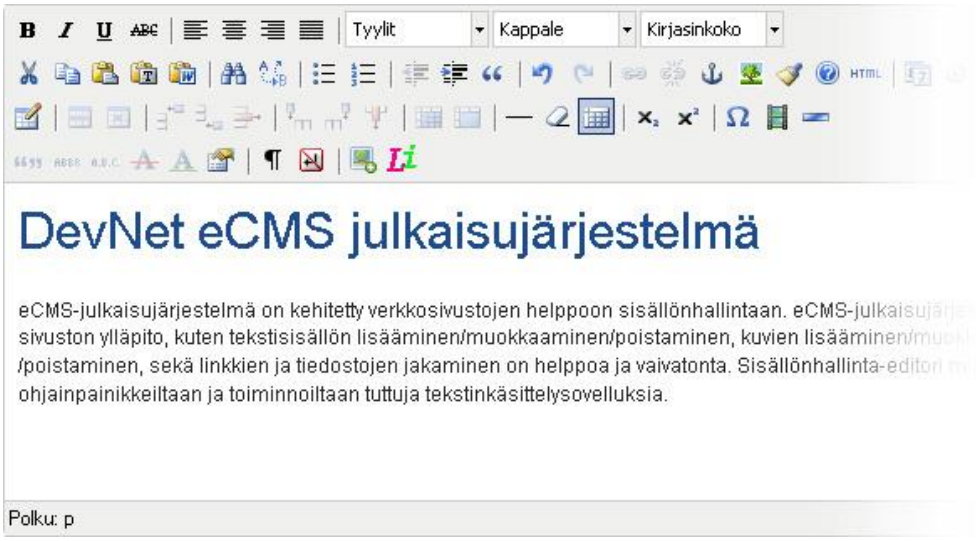
Suosittuja avoimeen lähdekoodiin perustuvia järjestelmiä ovat esimerkiksi WordPress sekä Joomla! (Ubinet 2015). WordPress kehitettiin alun perin blogien tekemiseen mutta myöhemmin sitä on alettu käyttämään myös perinteisissä verkkosivuissa. WordPress käyttää tietokantaa, johon kaikkia julkaisut ja artikkelit tallennetaan. WordPressissä on myös paljon valmiita sivupohjia, joten sivuston tekeminen on melko yksinkertaista (kuvio 5). (WordPress 2015b.)



KUVIO 5. WordPress-julkaisujärjestelmän ohjausnäkymä (WordPress 2015a)

Joomla! on toiminnoiltaan hyvin samankaltainen kuin WordPress mutta vaatii hieman enemmän teknistä osaamista kuin WordPress. Joomla on myös ilmainen ja sisältää hieman kilpailijaansa vähemmän valmiita lisäosia ja teemoja. Joomlaa suositellaan käytettäväksi erityisesti verkkokaupoissa sekä sosiaalisen median sivustoissa. (Mening 2013.)

Suljetun lähdekoodin julkaisujärjestelmästä esimerkkinä on DevNet Oy:n oma eCMS-julkaisujärjestelmä, jota käyttämällä asiakas pääsee muokkaamaan verkkosivun sisältöä web-pohjaisen hallintatyökalun avulla (kuvio 6). eCMS-julkaisujärjestelmä hoitaa myös suuren osan teknisestä hakukoneoptimoinnista, mutta käyttäjän täytyy huolehtia sisältöön liittyvästä optimoinnista. (DevNet 2015b.)



KUVIO 6. eCMS-julkaisujärjestelmä (WMHost 2015)

3.6 Tietokoneelle tehty verkkosivu mobiililaitteella

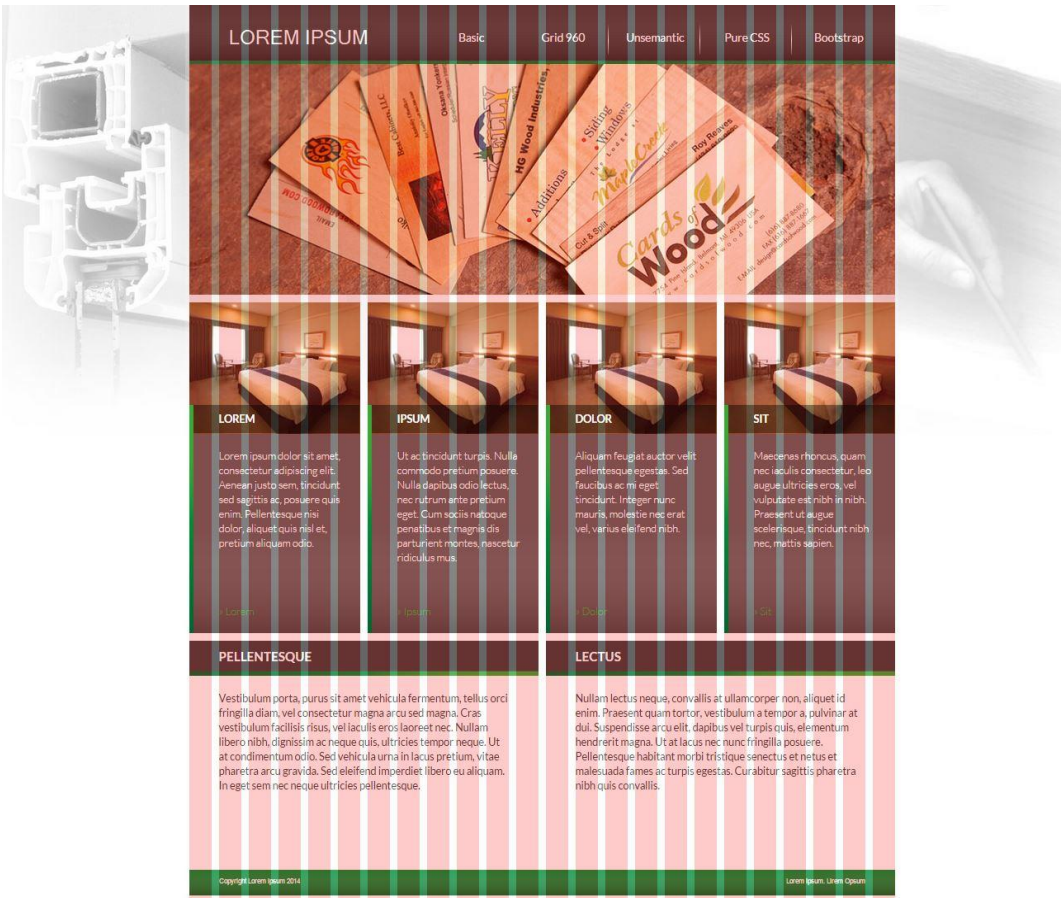
Erityisesti yrityksille on tärkeää, että heidän verkkosivunsa on tehty mobiililaitteilla toimivaksi. Yhdeksän kymmenestä mobiilikäyttäjästä on ilmoittanut poistuvansa yrityksen sivuilta, mikäli ne eivät toimi kunnolla mobiililaitteella. Tämä tarkoittaa käytännössä sitä, että yritykset menettävät asiakkaita, koska eivät edusta itseään toivotulla tavalla internetissä. (Mobiilimarkkinointi Routa Oy 2015.)

Verkkopalveluiden toimimattomuus mobiilissa tarkoittaa sitä, etteivät ne skaalaudu näytön koon mukaan (kuvio 7). Skaalautumattomuuden vuoksi käyttäjä joutuu vierittämään sivua runsaasti, minkä vuoksi sivun käytettävyys kärsii (Lerssi 2014). Skaalautumisongelmien välttämiseksi käytetään esimerkiksi 960 Grid System -frameworkia, jota esimerkiksi DevNet Oy käyttää verkkosivujen suunnittelussa ja työstämisessä.

KUVIO 7. HTML5-verkkosivu älypuhelimen näytöltä

3.7 960 Grid System

960 Grid System (lyh. 960.gs) on Nathan Smithin kehittämä framework, jota käytetään web-sivujen asettelussa. 960.gs:n avulla verkkosivu jaetaan joko 12, 16 tai 24 sarakkeeseen, ja nimensä mukaisesti kaikissa kolmessa tapauksessa sarakkeiden yhteisleveydeksi tulee 960 pikseliä (kuvio 8). Frameworkin tarkoituksena on, että verkkosivun leveys pysyy koko näytön levyisenä näyttökoosta huolimatta. Sarakkeiden tarkoituksena on helpottaa tekstien ja kuvien asettelussa. (Smith 2015a.)



KUVIO 8. 24-sarakkeinen mallipohja demosivun päällä

Frameworkissa sivuston rakenne suunnitellaan container-luokan sisälle. Container-luokka määrittelee myös sen, mihin sarakemäärään sivun rakenne tehdään (kuvio 9). Sivun ollessa esimerkiksi kolmepalstainen lisätään ensimmäiseen palstaan alpha-luokka ja kolmanteen palstaan omega-luokka, sillä muuten sarakkeiden asettelu toimii väärin, mikä rikkoo sivun rakenteen. Sisällön kuten tekstin alkamis- ja loppumispisteeseen voidaan vaikuttaa prefix- ja suffix-luokilla, jotka lisäävät tyhjän tilan tekstin alkuun tai loppuun. (Roydee 2010.)

```
<div class="container_24">
    <div class="grid_8 alpha">33 prosentin levyinen</div>
    <div class="grid_11 prefix_1">
        50 prosentin levyinen, palstan alussa yhden sarakkeen
        kokoinen tyhjä tila.
    </div>
    <div class="grid_4 omega">17 prosentin levyinen</div>
</div>
```

KUVIO 9. Esimerkki 24-sarakkeisesta rakennejaosta

Frameworkin käyttö nopeuttaa verkkosivujen suunnittelemista sekä koodaamista, sillä sarakkeiden avulla esimerkiksi web-ohjelmoijan on yksinkertaisempaa päätellä, mistä graafikko haluaa tekstin tarkalleen alkavan ja mihin päättyvän. 960.gs mahdollistaa myös monimutkaisten sivujen tekemisen erilevyisillä sarakkeillaan. (Way 2009.)

3.8 960 Grid System -sivu mobiililaitteella

Mobillilaitteella katsottaessa verkkosivun ulkoasu näyttää samalta kuin tietokoneen näytöltä katsottaessa. 960.gs:n avulla verkkosivu skaalautuu koko näytön levyiseksi, mutta ongelmaksi muodostuu sisällön pieneneminen (kuvio 10). Esimerkiksi viiden tuuman näytöllä (360 pikseliä leveä) frameworkia käyttävän sivun sisältö pienenee lähes kolme kertaa pienemmäksi, minkä vuoksi sivustoa pitää suurentaa (zoomata).

KUVIO 10. 960 Grid System -demosivu älypuhelimen näytöltä

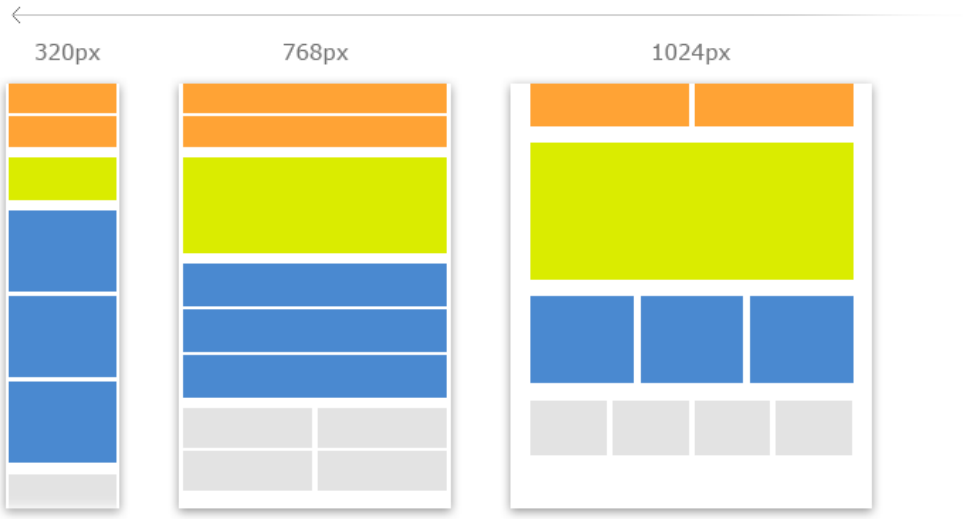
960.gs ratkaisee joitain skaalautuvuuteen liittyviä ongelmia, mutta se ei silti korjaa mobiililaitteella verkkosivujen käyttökokemusta riittävästi.

Skaalautuva ja ruudulle optimoituva verkkopalvelu on mahdollista tehdä responsiivisen verkkosuunnittelun avulla. (Hornor 2013.)

4 RESPONSIIVINEN VERKKOSUUNNITTELU

4.1 Määritelmä

Responsiivisen verkkosuunnittelun avulla verkkosivu mukautuu oikean näköiseksi eri resoluutioiden ja eri näytön kokojen mukaan (kuvio 11). Tämä tapahtuu CSS3:n media queryjen sekä CSS-tyylitiedostojen avulla, joita käyttämällä on mahdollista vaihtaa sivun kokoa aina tietokoneen suuresta näytöstä pienimmänkin älypuhelimen näytön kokoiseksi (Quinta Group 2015). Esimerkiksi älypuhelimen näytöllä näkyvä yksipalstainen sisältöelementti voi näkyä tabletin näytöllä kaksipalstaisena sisältöelementtinä (LePage 2015). Tarkoituksena on luoda käyttäjälle verkkosivusta mahdollisimman optimaalinen käyttökokemus, näytön koosta tai päätelaitteesta riippumatta. (Quinta Group 2015.)



KUVIO 11. Responsiivinen nettisivu kolmella erikokoisella näytöllä (Quinta Group 2015)

Responsiivisen verkkosivusuunnittelun kehittäjänä pidetään Ethan Marcottea. Hän kirjoitti vuonna 2010 artikkelin Responsive Web Design, jossa painotettiin, että mobiililaitteet tulevat yleistymään merkityksellisen paljon muutaman seuraavan vuoden ajan. Marcotte on tiivistänyt

responsiivisuuden kolmeen käsitteeseen: ”fluid grids, flexible images and media queries”, joista vain jälkimmäinen käsite oli todella jotain uutta. (Korpela 2012.)

Responsiivisessa verkkosuunnittelussa käytetään usein mobile first -ajatusta, jonka mukaan ei ole aina järkevintä suunnitella ensin työpöytänäkymää ja karsia siitä mobiilinäkymä, vaan tuottaa pääasiat mobiilinäkymälle sopivaksi ja tämän jälkeen laajentaa siitä toimiva kokonaisuus työpöytänäkymään. Mobile first -ajatuksen kautta sivuston suunnittelu ja rakenne tapahtuvat eri tavalla kuin normaalisti ja tällöin sivusto saattaa saada paljon selkeämmän ja toimivamman rakenteen sekä mobiililaitteella ja tietokoneella käytettäessä. (Leiniö 2012.)

4.2 Kilpailijat

Responsiivisen verkkosuunnittelun lisäksi mobiilisivujen tekemiseen on kolme vaihtoehtoa:

1. Luoda erillinen mobiilisivu verkkosivuista.
2. Luoda erillinen natiivi-sovellus.
3. Luoda erillinen hybridisovellus.

Erillisen mobiilisivun tekeminen aiheuttaa runsaasti lisätyötä, sillä ylläpidettäväksi tulee 2 erillistä sivustoa, mobiili sekä normaali. Sivustoilla saattaa myös olla täysin eri lähdekoodi, minkä vuoksi niitä ei pysty hyödyntämään keskenään (Leiniö 2012). Mikäli verkkosivun tekemiseen käytetään sisällönhallintajärjestelmää, on responsiivinen verkkosivu parempi vaihtoehto, sillä sen päivittämiseen ei tarvitse kuin yhden järjestelmän ylläpidon. (Dainow 2014.)

Natiivi-sovelluksen haasteena on, että joka käyttöjärjestelmälle täytyy tehdä oma sovelluksensa. Esimerkiksi Androidille ja Windows Phonelle tehtävät sovellukset eivät ole yhteensopivia. Sovelluksiin tehtävät muutokset on myös tehtävä jokaisen alustan sovelluksiin erikseen, mikä vaatii runsaasti osaamista. (Leiniö 2012.)

Hybridisovellus on yhdistelmä HTML5-koodia ja natiivi-koodia. Tämä tarkoittaa sitä, että käyttöliittymä tai osa siitä tehdään käyttämällä HTML5-koodia. Koodi pakataan sovellukseksi käyttämällä esimerkiksi PhoneGap frameworkia. Natiivi-sovellukseen verrattuna hybridisovelluksen etuna on, että se voidaan kääntää yksinkertaisesti eri mobiilialustoille sopivaksi. Hybridisovelluksen heikkoutena on, että mobiilisovelluksen tavoin joudutaan ylläpitämään 2:ta sivustoa. Lisäksi hybridisovelluksen koodiin joudutaan usein tekemään korjauksia, jotta sovellus kääntyy toimivaksi kaikilla tarvittavilla mobiilialustoilla. (Riippi 2013.)

4.3 Responsiivisen verkkosuunnittelun käyttö

4.3.1 Media query

Media query on CSS3:n osa, jonka avulla voi tehdä kohdistettua CSS-tyylittelyä esimerkiksi näytön koon tai orientaation mukaan. Media queryn tarkoituksena on optimoida sivu hyvännäköiseksi millä näyttölaitteella tahansa (van Hove 2015). Media queryn avulla lajitellaan laitteet ryhmiin, joista jokaiselle tehdään omat asettelunsa. Jokaisen asettelun sisällä noudatetaan joustavaan sommitteluun liittyviä periaatteita. (Korpela 2012.)

Media queryn käytön kannalta breakpoint on oleellinen käsite. Breakpoint on niin sanottu taitekohta, jonka jälkeen sivuston rakenne tai tyyli muuttuu. Breakpoint voidaan määrittää joko CSS-tiedostossa (kuvio 12) tai HTML-tiedoston Head-elementissä (kuvio 13). Esimerkiksi tabletin ruudulla voidaan näyttää enemmän informaatiota asioista kuin mobiilinäytöllä, jolloin mobiililaitteen ja tabletin määritetyssä breakpointissa epäoleellinen informaatio piilotetaan näkyvistä (LePage 2014a). Sisällön piilottamista käydään enemmän läpi luvussa 4.3.5

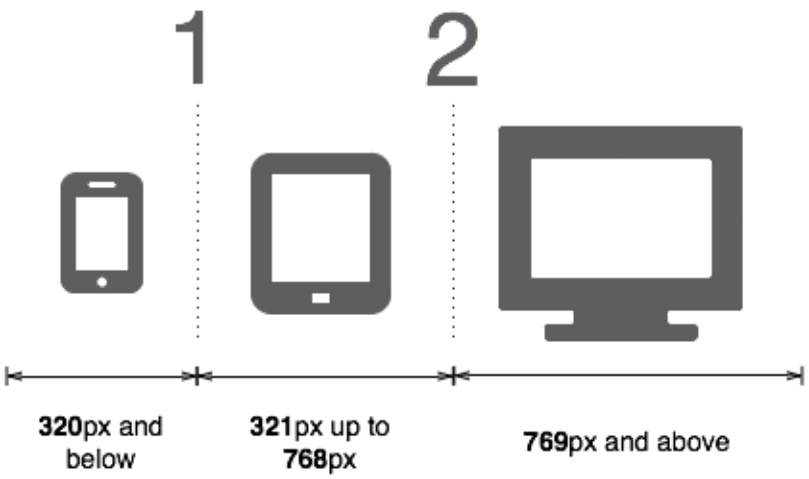
```
#alue1, #alue2 {
    width: 100%;
    float: right;
}

@media (min-width: 800px) {
    #alue1, #alue2 {
        width: 50%;
    }
}
```

KUVIO 12. Esimerkki Media queryn käytöstä CSS-tiedoston sisällä

```
<link rel="stylesheet" media="(max-width: 640px)" href="max-640px.css">
<link rel="stylesheet" media="(min-width: 640px)" href="min-640px.css">
<link rel="stylesheet" media="(orientation: portrait)" href="portrait.css">
<link rel="stylesheet" media="(orientation: landscape)" href="landscape.css">
```

KUVIO 13. Näyttökokokohtaiset CSS-tiedostomäärittelyt Head-elementissä (LePage 2014a)

Breakpointteja käytetään pääasiassa vain pakollisiin tarkoituksiin, sillä mitä enemmän on taitekohtia, sitä enemmän on tyyliteltävää. Breakpointteja on kuitenkin yleensä vähintään kaksi, joista toinen erottaa älypuhelimen ja tabletin näytön ja toinen tabletin ja tietokoneen näytön (kuvio 14) (LePage 2014b.)



KUVIO 14. Älypuhelimen, tabletin ja tietokoneen breakpointit (Ford 2013)

Kaikki internet-selaimet eivät tue media querya kokonaisuutena, mutta selaimet yrittävät silti suorittaa sen. Tämä johtuu siitä, että media queryt on pääsääntöisesti suunniteltu niin, että ne toimivat kaikissa moderneissa mobiiliselaimissa (Korpela 2012). Media queryyn voidaan kuitenkin lisätä erillinen esto ”only” (kuvio 15). Eston avulla vanhat selaimet ohittavat media queryn määrittelyn kokonaan mutta uudemmat tukevat selaimet pystyvät silti suorittamaan määrittelyn (Knight 2011). Vanhemmat selaimet voidaan kuitenkin ottaa huomioon sopivien JavaScript-koodien avulla. Skriptin tarkoituksena on saada selain lukemaan ja jopa toteuttamaan CSS-koodissa olevat määritykset. Esimerkkinä JavaScript-kirjastosta on Respond.js, joka pystyy suorittamaan esimerkiksi max-width- ja min-width-rajoitukset. (Korpela 2012.)

```
@media="only screen and (max-width: 600px)" {
}
```

KUVIO 15. Eston käyttö media queryssä

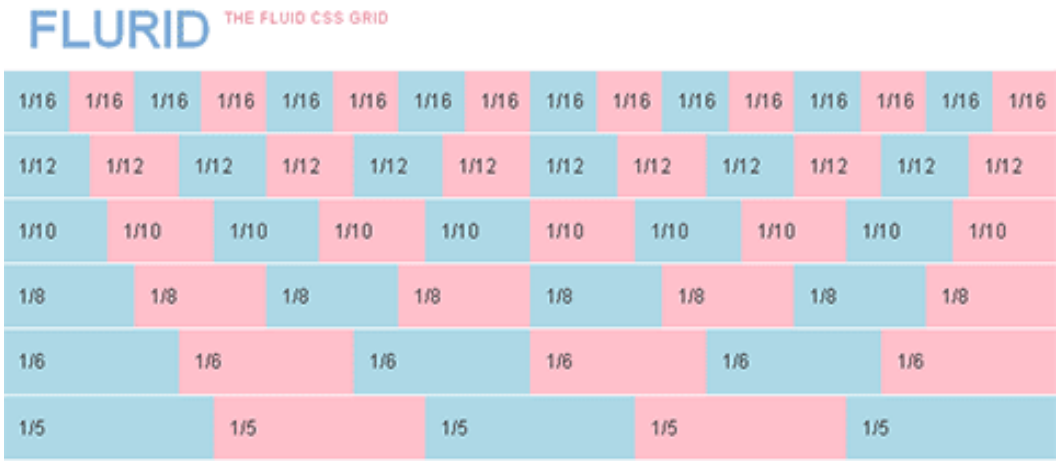
4.3.2 Viewport

Viewport on HTML-tiedoston meta-tietoihin laitettava tagi, jolla voidaan määrittää, miltä verkkosivu näyttää mobiililaitteen selaimessa. Viewportin avulla voidaan säätää sivun leveyttä, korkeutta ja skaalautuvuutta. Viewport toimii lähes kaikissa moderneissa mobiiliselaimissa. (Ala-Äijälä 2012.)

Viewportin leveys-parametrille voidaan asettaa numeerinen pikseliarvo, mutta sen arvoksi voidaan asettaa myös ”width = device-width”, jolloin sivu näkyy koko näytön levyisenä eli skaalautuvana. Skaalautuvuudelle voidaan myös asettaa rajat tai se voidaan estää kokonaan. (Ala-Äijälä 2012.)

4.3.3 Fluid Grid

Fluid Gridin eli joustavan taittopohjan idea on samankaltainen kuin 960.gs:ssä eli sivu jaetaan sarakkakkeisiin. 960.gs:n vakiomittaisten sarakkeiden sijaan Fluid Gridissä käytetään suhteellisen mittaisia sarakkeita (kuvio 16). (Knight 2009.)



KUVIO 16. Esimerkkitaulu suhteellisesta jakamisesta (Flurid 2014)

Fluid Gridin lisäksi voidaan käyttää Elastic Gridiä eli joustavaa taittopohjaa sekä Fixed Gridiä eli määrätyn kokoista taittopohjaa. Elastic Gridin idea on samanlainen kuin Fluid Gridin: se skaalautuu ja mukautuu samalla tavalla, mutta vain tiettyyn rajaan saakka. Elastic Gridillä on maksimi- ja miniarvo, jonka jälkeen taittopohja ei enää veny. Tämän ansiosta verkkosivun sisältö ei voi venyä liian suureksi, vaikka näyttökoko olisikin suuri.

Fixed Grid toimii hyvin samalla tavalla kuin 960.gs-framework. Taittopohja pysyy tietyn kokoisena, eikä se skaalaudu sisältönsä tai näytön koon mukaan. Fixed Gridin avulla voidaan kuitenkin käyttää breakpointteja, jolloin sivuston ulkoasu muuttuu tiettyjen taitekohtien mukaan. (Knight 2009.)

4.3.4 Fluid Images

Fluid Images eli joustavat kuvat ovat kuvia, jotka skaalautuvat näytön koon mukaan. Responsiivisessa verkkosuunnittelussa on tärkeää, että kuvat

ovat skaalautuvia, sillä muuten kuvat voivat aiheuttaa ongelmia käyttöliittymän rakenteessa ja hajottaa sen. Skaalautuvuus asetetaan CSS-tiedostossa, jossa kuva-attribuutin maksimileveydeksi asetetaan 100 prosenttia. Kyseinen määritys ei toimi Internet Explorer-selaimella, mutta se on mahdollista korjata laittamalla kuva-attribuutin kokonaisleveydeksi 100 prosenttia (kuvio 17). (Leiniö 2012.)

```
img{
     max-width: 100%;
     width: 100%;
}
```

KUVIO 17. Skaalautuvien kuvien määritykset CSS-tiedostossa

4.3.5 Sisällön piilottaminen

Mobiilinäytölle ei mahdu niin paljon sisältöä kuin esimerkiksi tietokoneen näytölle, vaikka elementtejä pienentäisi kuinka paljon. Ongelma voidaan ratkaista sijoittamalla sisältö uudelleen mobiilisivulle mutta tämä saattaa pidentää sivua huomattavasti. (Leiniö 2012.)

Toinen vaihtoehto on piilottaa vähemmän tärkeät elementit kokonaan mobiilisivusta (kuvio 18). Elementtien piilottaminen tapahtuu CSS:n avulla, ja se tapahtuu joko käyttämällä ”visibility: hidden-” tai ”display = none” -määritettä. Ensimmäinen vaihtoehto ainoastaan piilottaa sisällön, mutta jälkimmäinen myös poistaa sisällön. Mikäli sisältöä ei poisteta, sisällön kohdalle jää sille varattu tyhjä tila. (Leiniö 2012.)

KUVIO 18. Vähemmän tärkeän elementin piilottaminen mobiilisivulta (Weil 2014)

4.3.6 Yleisiä responsiivisuuteen vaikuttavia määrityksiä

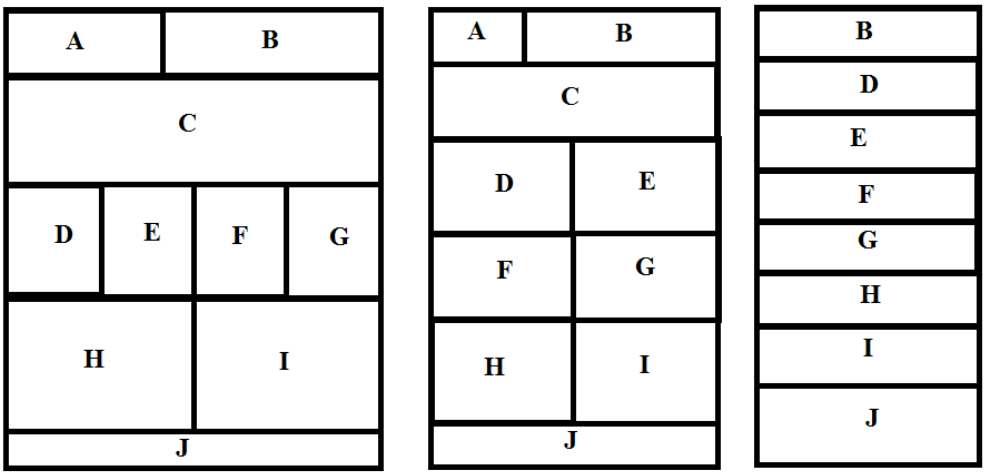
Responsiivinen suunnittelu on kokonaisuudessaan laajempi käsite kuin aikaisemmin on käsitelty. On tärkeää, että sivun kokonaisasettelu on suhteessa laitteen kokoon responsiivinen. Lisäksi tulee huomoida, että painikkeet sekä linkit ovat riittävän suuret, jotta niitä voidaan käyttää kosketusnäytöllä. Kuvien tulee olla mahdollisimman pienikokoisia, sillä on huomioitava myös hidasyhteyksisten laitteiden käyttäjät. Lisäksi ylimääräisten ja tarpeettomien kuvien käyttöä tulee välttää, sillä ne vievät runsaasti ylimääräistä tilaa sisällöltä näytöllä. (Korpela 2012.)

Sisällön tulee myös olla tiivis ja tarpeellinen. Mikäli vähemmän tärkeän sisällön haluaa pitää mobiilisivulla, se kannattaa piilottaa niin, että sen voi tarvittaessa saada näkyviin esimerkiksi ”lue lisää” -painikkeen avulla. Sivustossa tulee ottaa myös huomioon fonttivalikoima. Mobiililaitteen fonttivalikoima on melko suppea, ja tämän vuoksi fonttien määrää voi lisätä käyttämällä ladattavia fontteja, kuten esimerkiksi Googlen Web-fontteja. Lisäksi tekstin koon tulee olla riittävän isoa, jottei sivua tarvitse suurentaa jatkuvasti. (Korpela 2012.)

5 RESPONSIIVISET FRAMEWORKIT

Tässä luvussa käsittellään kolmea eri frameworkia, jotka on valittu vertailuun keskenään. Unsemantic on 960.gs:n kehittäjän tekemä responsiivinen framework, ja tämän vuoksi se on valittu yhdeksi kolmesta (Smith 2015d). Toisena frameworkina käsittellään Pure.css-frameworkia, joka on pienikokoinen ja erittäin kevyt framework (Pure 2014). Kolmantena frameworkina käsitellään Bootstrapia, joka on maailman käytetyin responsiivinen framework. (Bootstrap 2015.)

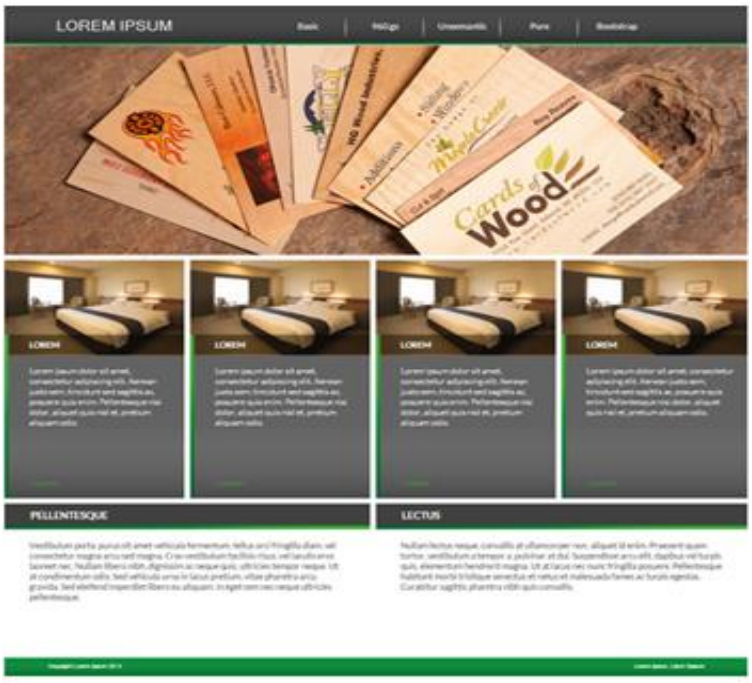
Jokaista frameworkia varten on tehty oma demosivu, ja niiden rakenne pyritään pitämään samana (kuvio 19). Koska alkuperäinen demosivu on suunniteltu ja tehty 24-sarakkeiseen 960.gs:n pohjaan, joudutaan joissain tapauksissa rakenteessa poikkeamaan hieman, sillä kaikissa frameworkeissa ei ole käytössä 24-sarakkeista jakoa. Mobiilisivun rakenteesta on poistettu kohdat A ja C, sillä tarkoituksena on tuoda näytön koon pienuuden vuoksi sisältö paremmin esille. A-kohdassa on paikka logolle ja C-kohdassa paikka vierityspalkille (slider) tai kuvalle.



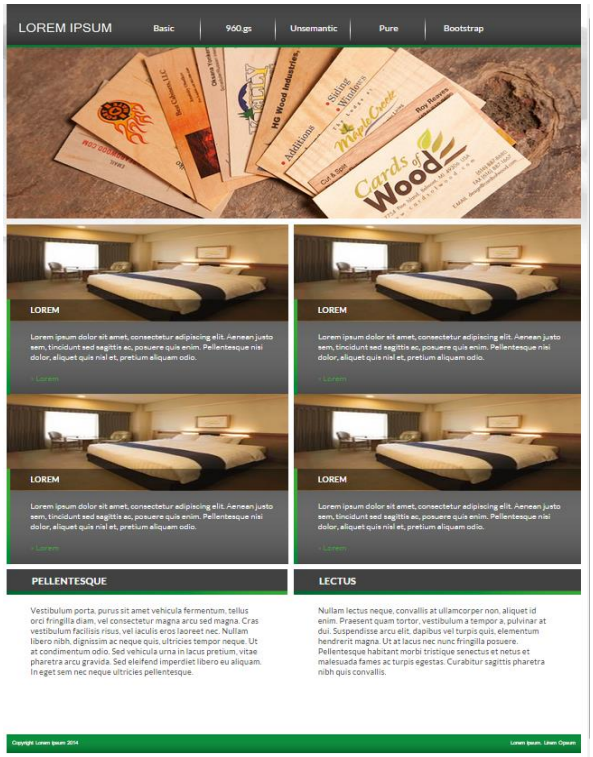
KUVIO 19. Demosivuston rakennemallit

Demosivujen on tarkotus mukautua tietokoneen (kuvio 20), tabletin (kuvio 21) ja älypuhelimen näytölle sopivaksi (kuvio 22). Jokaisessa sivussa käytetään siis ainoastaan kahta breakpointia: yhtä mobiilinäytön ja tabletin näytön ja toista tabletin näytön ja tietokoneen näytön välille. Tarkoituksena

on myös käsitellä, viekö jonkin frameworkin käyttö enemmän aikaa kuin toisen.



KUVIO 20. Demosivu tietokoneen näytölle optimoituna



KUVIO 21. Demosivu tabletin näytölle optimoituna

KUVIO 22. Demosivu älypuhelimen näytölle optimoituna

5.1 Unsemantic

5.1.1 Kuvaus

Unsemantic on Nathan Smithin kehittämä responsiivinen framework, joka on 960.gs:n seuraaja. Se toimii samankaltaisesti kuin 960.gs, mutta sarakkeiden lukumäärän sijaan se noudattaa prosentuaalista jaottelua (kuvio 23). Jaottelussa sarakkeiden arvot menevät viiden kertotaululla viidestä sataan asti. Unsemantic ymmärtää myös arvot 33 ja 66 eli yksi kolmasosaa ja kaksi kolmasosaa. Toinen eroavaisuus 960.gs:n ja Unsemanticin välillä on gridin leveys. 960.gs:n 960 pikselin sijaan Unsemanticin gridin leveys on 1180 pikseliä. (Smith 2015c.)

```
<div class="grid-container"> <!-- 100 prosentia leveydestä -->
    <div class="grid-35">
        35 prosentin levyinen
    </div>

    <div class="grid-25">
        25 prosentin levyinen
    </div>

    <div class="grid-40">
        40 prosentin levyinen
    </div>
</div>
```

KUVIO 23. Esimerkki Unsemanticin käytöstä

Unsemanticin avulla voidaan näyttää tietty elementti eri kokoisena eri näyttölaitteella. Sen avulla voidaan myös piilottaa elementtejä tai näyttää tietty elementti vain tietyllä näyttökoolla (kuvio 24). (Smith 2015b.)

```
<div class="grid-container">
    <div class="mobile-grid-100 grid-50">
        Mobiilinäytöllä 100 prosentin levyinen.
        Tietokoneen näytöllä 50 prosentin levyinen.

        <span class="hide-on-tablet">
            Piilotettu tabletin näytöltä
        </span>
    </div>
</div>
```

KUVIO 24. Esimerkki eri kokoisista elementeistä sekä elementin piilottamisesta

5.1.2 Unsemanticin käyttö demosivulla

Unsemantic vie tilaa tietokoneelta alle yhden megantavun, mikä on kohtalainen koko frameworkille. Unsemantic sisältää valtavan määrän tiedostoja, joista pelkästään CSS-tyylitiedostoja on 34 kappaletta. Dokumentoinnin avulla löytyy lopulta oikeat tyylitiedostot, jotka ovat reset.css ja unsemantic-grid-responsive-tablet.css. Jälkimmäinen tyylitiedosto sisältää breakpointit mobiililaitteen ja tabletin välille sekä tabletin ja tietokoneen näytön välille.

Verkkosivujen tekeminen aloitettiin tietokonenäkymästä. Koska alkuperäiset sivut on tehty 960 pikseliä leveällä gridillä, tuli sivusta hieman leveämpi, sillä Unsemanticin gridin leveys on 1180 pikseliä. Suurimmaksi ongelmaksi koitui ylemmässä sisältöelementissä olevien neljän palstan välit. Alunperin välit tulivat suoraan 960.gs:n tyylimäärityksistä, mutta Unsemanticin määrityksissä kyseiset välit olivat paljon leveämmät, ja tämän vuoksi niiden käyttäminen ei sopinut ulkoasuun. Välien sijaan palstojen oikeaan reunaan asetettiin oikean levyinen valkoinen kehys (border), jonka ansiosta sivun ulkoasu vastasi alkuperäistä.

5.2 Pure

5.2.1 Kuvaus

Pure on erittäin pienikokoinen responsiivinen framework. Puren tarkoituksena on sisältää vain kaikki oleelliset CSS-määritykset, koska Puren tekijöiden mukaan on helpompaa tehdä uusi CSS-määritys kuin päällekirjoittaa olemassa oleva. Myös Pureen liittyvät tyylit on tehty niin, että ne on tarvittaessa helppo päällekirjoittaa (Pure 2014b). Purea käytetään yhdessä Normalize.css:n kanssa. Normalize.css on tiedosto, joka renderöi kaikki elementit johdonmukaisiksi ja modernien standardien mukaisiksi. (Gallagher & Neal 2015.)

Puren grid on mahdollista jakaa joko 5 tai 24 osaan. Gridin jaottelu tapahtuu prosentien sijaan murtoluvuilla. Esimerkiksi Puren merkintä 1-4 tarkoittaa 25 % näytön leveydestä (kuvio 25). (Pure 2014b.)

```
<div class="pure-g">
    <div class="pure-u-1-4">25 prosentia näytön leveydestä</div>
    <div class="pure-u-1-2">50 prosentia näytön leveydestä</div>
    <div class="pure-u-1-4">25 prosentia näytön leveydestä</div>
</div>
```

KUVIO 25. Esimerkki Puren käytöstä

Puren CSS-tiedosto sisältää valmiita tyylimäärityksiä esimerkiksi painikkeille, taulokoille ja lomakkeille (kuvio 26) (Pure 2014a). Valmiit

määritykset ovat hyödyksi esimerkiksi graafikolle, joka suunnittelee sivuston ulkoasua, sillä hän voi valmiiksi valita oikean näköisen elementin ja näin vähentää omaa sekä koodaajan työmäärää.

```
<a class="pure-button" href="#">A Pure Button</a>
<button class="pure-button">A Pure Button</button>
```

KUVIO 26. Puren määritykset kahdelle napille (Pure 2014)

5.2.2 Puren käyttö demosivulla

Purea ei tarvitse ladata tietokoneelle lainkaan, sillä css-tiedoston voi sisällyttää HTML-tiedostoon suoraan verkkosivulta. Puren tiedostot voi myös ladata koneelleen. Tiedostot vievät hieman yli 200 kilotavua tilaa, ja ne ovat suurimmaksi osaksi CSS-tyylitiedostoja lomake- ja taulukkomäärityksiä varten. Vaikka CSS-tiedostoja on lähes yhtä monta kuin Unsemanticissa, on tiedostot nimetty paljon selkeämmin ja ymmärrettävämmin. Puresta käytetään siis vain pure-min.css- sekä normalize.css-tiedostoa.

Pure perustuu mobile first -ideaan, minkä vuoksi sivujen tekeminen aloitettiin mobiilinäkymästä. Puren gridin koko on myös koko ruudun levyinen, minkä vuoksi fontteja ja kuvia joutui suurentamaan, jotta ne näkyisivät hyvin. Puressa on mahdollisuus käyttää kolmea breakpointia, mutta suurin näyttökoko laitettiin vastaamaan tietokoneen näytön kokoa.

Puren gridin jaottelu noudattaa 24 sarakkeen jakoa, minkä vuoksi palstojen jako toimi tietokoneen näytöllä samalla tavalla kuin 960 Grid Systemissa. Mobiilisivun tekemisen jälkeen oli yksinkertaista laajentaa sivu vastaamaan tabletille ja tietokoneen näytölle suunniteltua rakennetta. Unsemanticin lisäksi myös Puren ylemmän sisältöelementin välien tekemiseen käytettiin kehyksiä.

Purella sivujen tekeminen oli nopeampaa kuin Unsemanticilla. Mobiilisivun laajentaminen suuremmaksi vei huomattavasti vähemmän aikaa, sillä yhteensopivuusongelmilta vältyttiin.

5.3 Bootstrap

5.3.1 Kuvaus

Bootstrap on maailman suosituin responsiivinen framework. Bootstrap perustuu mobile first -ideaan, eli ensin suunnitellaan verkkosivu mobiililaitteelle ja tämän jälkeen laajennetaan se tietokoneen näytölle sopivaksi. Bootstrap on avoimeen lähdekoodiin perustuva julkaisu, ja sitä ylläpidetään Githubin kautta. Puren lisäksi myös Bootstrap käyttää Normalize.css:ää. (Bootstrap 2015).

Bootstrapin grid on jaoteltu 12 sarakkeen mukaisesti, ja sen leveys on maksimissaan 1170 pikseliä. Bootstrapissa on neljä kokoluokkaa aina erittäin pienestä suureen näyttökokoon. Jokaiselle kokoluokalle on annettu oma lyhenteensä, ja niitä käytetään sarakkeiden lisäksi nettisivujen teossa (kuvio 27). (Bootstrap 2015.)

```
<div class="row">
    <div class="col-xs-6 col-md-3">Mobiilissa 50%, Tabletilla 25%</div>
    <div class="col-xs-4">Kaiken kokoisilla näytöillä 33% leveydestä</div>
    <div class="col-xs-12 col-lg-2">Mobiililaitteella 100%, suurella näytöllä 16%</div>
</div>
```

KUVIO 27. Esimerkki Bootstrapin käytöstä

Kuten Puressa myös Bootstrapissa on valmiita tyylimäärityksiä taulukoille, lomakkeille ja painikkeille. Frameworkiin on myös tehty runsaasti muita tyylimäärityksiä, minkä vuoksi netissä oleva dokumentointi onkin erittäin laaja (Bootstrap 2015). Runsaat tyylimääritykset ovat osittain myös haitta, sillä mahdollisissa päällekirjoitustilanteissa tulee todella pitää huoli siitä, että oikeat tyylit ovat käytössä

5.3.2 Bootstrapin käyttö demosivulla

Bootstrap on valituista frameworkeista kaikista suurin, sillä sen koko on tietokoneella vähän yli yhden megantavun, mikä ei kuitenkaan ole liian paljon web-ohjelmoinnissa. Tiedostomäärällisesti Bootstrap on taas kaikista pienin, sillä se sisältää ainoastaan neljä CSS-tiedostoa, joista kaksi on pienennettyjä ja kompressoituja versioita alkuperäisistä tiedostoista. Puren tapaan myös Bootstrapin voi sisällyttää nettisivulle internetistä. Bootstrapin tiedostoista käytetään bootstrap.css- sekä normalize.css-tyylitiedostoja.

Bootstrap on toiminnaltaan hyvin samankaltainen kun Pure. Puren lisäksi myös Bootstrap perustuu mobile first -ajatukseen, ja täten verkkosivujen tekeminen aloitettiin mobiilisivusta. Nettisivujen laajentaminen onnistui helposti ja myös tässä tapauksessa suurin näytön koko laitettiin vastaamaan tietokoneen näytön kokoa. Suurin ero sivun tekemisessä Pureen oli, että Bootstrap noudattaa ainoastaan 12-sarakkeista jakoa, kun taas Pure noudattaa 24-sarakkeista jakoa. Bootstrap tarvitsee myös toimiakseen JavaScriptin mutta Puren käyttö ei sitä vaadi.

Puren ja Bootstrapin välillä ei ollut juurikaan eroa nettisivujen tekemisessä. Kyseisen nettisivun tekemiseen kului vähän vähemmän aikaa Purella, mutta todennäköisenä syynä on yhtenevä sarakejako 960.gs:n kanssa.

6 FRAMEWORKIEN VÄLINEN VERTAILU

Jokainen framework toimi niin, että niillä saatiin onnistuneesti tehtyä verkkosivut, jotka mukautuvat tarvittavien näytön kokojen mukaisesti. Frameworkien väliseen vertailuun on käytetty taulukkoa, jossa on tietyt kriteerit hyvälle ja toimivalle frameworkille (taulukko 1).

TAULUKKO 1. Kolmen responsiivisen frameworkin vertailu

	Unsemantic	Pure	Bootstrap
Helppo käyttöönotto	X	✓	✓
Fyysinen koko	✓	✓	✓
Yksinkertainen käytettävyys	✓	✓	✓
Monipuolisuus	X	✓	✓
Valmiita määrityksiä elementeille*	X	✓	✓
Kaikki tarvittava yhdessa CSS-tiedostossa	X	X	✓
Ei vaadi JavaScriptia	✓	✓	X
Mobile first	X	✓	✓

* Valikot, taulukot, painikkeet..

Web-ohjelmoinnissa on tärkeää, että kaikki tarvittava on helposti saatavilla yhdestä ja samasta CSS-tiedostosta ja tämän vuoksi Bootstrap on paras vaihtoehto responsiiviseen verkkosuunnitteluun. Puren ainoana heikkoutena oli, että määritykset valmiille elementeille olivat useassa eri

CSS-tiedostossa. Unsemantic on hyvä perus framework, joka toimii samankaltaisesti kuin 960.gs. Unsemantic ei siitä huolimatta tarjoa samanlaista käyttömukavuutta kuin kaksi muuta frameworkia. Unsemantic on myös valikoimaltaan hieman suppea eikä tarjoa responsiivisuuden lisäksi juuri muuta.

Suunnittelijan näkökulmasta Bootstrap on myös hyvä vaihtoehto, sillä se tarjoaa valmiita elementtejä, joita graafikko voi käyttää nettisivujen ulkoasun suunnittelussa. Lisäksi mobiilisivun kunnollinen suunnitteleminen vähentää suunnitteluun ja työstämiseen käytettävää aikaa, sillä mobiilisivu on erittäin vika-altis.

960 Grid System-frameworkin korvaamiseen Bootstrap on erittäin hyvä vaihtoehto. Se on yksinkertainen käyttää, ja se noudattaa mobile first -ideaa, jonka ansioista verkkosivut toimivat hyvin mobiilissa. Lisäksi Bootstrap tarjoaa graafikolle valmiita malleja taulukoiden ja painikkeiden suunnitteluun, minkä vuoksi graafikon ja web-ohjelmoijan välinen yhteistyö helpottuu ja selkeytyy. Lisäksi modernin frameworkin käyttö saattaa kasvattaa yrityksen positiivista imagoa asiakkaan silmissä.

7 YHTEENVETO

Verkkosivuja luetaan nykyisin aina kun siihen on mahdollisuus. Perinteisen tietokoneen sijaan mobiililaitteet ja tabletit ovat vieneet suuren osan internetin käyttäjistä perinteisiltä sivuilta optimoiduille responsiivisille sivuille. Tämän vuoksi yhä useampi verkkopalvelu suunnitellaan alusta lähtien responsiiviseksi.

Responsiivisten frameworkien käyttö helpottaa erittäin paljon verkkosivujen tekemistä ja ylläpitoa. Erillisten web-mobiilisivujen sijaan responsiivinen verkkosivu tavoittaa kerralla tietokoneen, tabletin ja älypuhelimen käyttäjät. Responsiivisten verkkosivujen tekeminen ja suunnittelu vie hieman enemmän aikaa kuin perinteisten verkkosivujen tekemisen, mutta verratuna perinteisen verkkosivun ja web-mobilisivun suunnitteluun se on nopeampaa ja tehokkaampaa. Lisäksi yhden sivun ylläpitäminen vie vähemmän aikaa kuin kahden erillisen sivun.

Responsiivisia frameworkeja on satoja erilaisia, mutta kaikilla niillä on yksi perustarkoitus: tarjota käyttäjälle mahdollisimman hyvä käyttömukavuus käyttölaitteesta tai näytön koosta riippumatta. Opinnäytetyöhön valitut frameworkit ovat perustellusti ja harkitusti valittuja, mutta se ei tarkoita sitä, ettei jokin toinen framework voisi soveltua DevNet Oy:n käyttöön paremmin kuin vertailussa valittu Bootstrap. 960.gs:n tekniikka ei kuitenkaan enää riitä vastaamaan nykypäivän tarpeeseen, jossa verkkosivujen tulee toimia millä laitteella tahansa.

Bootstrap on maailman suosituin, erittäin toimiva ja jatkuvasti päivittyvä framework, minkä vuoksi sen käyttöön kannattaa tutustua. Bootstrap tarjoaa myös paljon valmiita elementtejä, joiden käyttö nopeuttaa sekä suunnittelijan ja web-ohjelmoijan työtä.

Responsiivisten verkkosivujen tulevaisuus näyttä erittäin vakaalta. Uudet responsiiviset frameworkit tuovat jatkuvasti jotain pientä uutta markkinoille, mutta jos HTML 5.1 ja 3D-tekniikka eivät tuo mitään tärkeää markkinoille, pysyy responsiivisen verkkosuunnitelun idea pitkään samana.

LÄHTEET

Ala-Harja, H. 2014a. Hakukoneoptimoinnin pikaopas [viitattu 24.3.2015]. Saatavissa: http://www.google-optimointi.com/pikaopas/

Ala-Harja, H. 2014b. Hakukoneoptimointi kasvattaa kävijämäärää [viitattu 24.3.2015]. Saatavissa: http://www.google-optimointi.com/

Ala-Äijälä, J. 2012. Mobiiliselaimet ruotuun viewport-tagilla [viitattu 14.3.2015]. Saatavissa: http://www.aucor.fi/blogi/mobiiliselaimet-ruotuun-viewport-tagilla/

Bootstrap. 2015. Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web [viitattu 16.3.2015]. Saatavissa: http://getbootstrap.com

Bootstrap. 2015. CSS [viitattu 16.3.2015]. Saatavissa: http://getbootstrap.com/css/

CSS Neuse. 2015. The history of CSS [viitattu 7.3.2015]. Saatavissa: http://www.cssneuse.net/the-history-of-css.php

Dainow, B. 2014. Responsive design vs. mobile websites: And the winner is... [viitattu 12.3.2015]. Saatavissa: http://www.imediaconnection.com/content/36704.asp#multiview

DevNet Oy. 2015a. DevNet lyhyesti [viitattu 7.3.2015]. Saatavissa: http://www.devnet.fi/yritys

DevNet Oy. 2015b. Kotisivut ja julkaisujärjestelmä [viitattu 3.4.2015]. Saatavissa: http://www.devnet.fi/palvelut/ohjelmistototeutukset/kotisivut-ja-julkaisujarjestelma

Ekonoja, A., Lahtonen, J. & Mäntylä J. 2011. WWW-sivujen käytettävyys ja esteettömyys – Luento 10 [viitattu 23.3.2015]. Saatavissa: http://appro.mit.jyu.fi/www/luennot/luento10/

Flurid. 2014. A Cross-Browser & Fluid CSS Grid – Flurid [viitattu 15.3.2015]. Saatavissa: http://www.webresourcesdepot.com/a-cross-browser-fluid-css-grid-flurid/

Fonecta. 2015. Hakusanamainonta ohjaa asiakkaan sivuillesi, kun palveluitasi tarvitaan [viitattu 29.3.2015]. Saatavissa: https://www.fonecta.fi/yrityksille/google-mainonta/

Ford, P. 2013. Break Points [viitattu 13.3.2015]. Saatavissa: https://idn.irise.com/display/LIB/Break+Points

Gallagher, N. & Neal, J. 2015. Normalize.css – A modern, HTML5-ready alternative to CSS resets [viitattu 16.3.2015]. Saatavissa: http://necolas.github.io/normalize.css/

Heinonen, P. 2015. Hyvät kotisivut vauhdittavat myyntiä! [viitattu 23.3.2015]. Saatavissa: http://www.obra.fi/kotisivujen_suunnittelu.html

Hornor, J. 2013. Build a Responsive Design using 960 grid [viitattu 11.3.2015]. Saatavissa: http://www.sitepoint.com/build-a-responsive-design-using-960-grid/

Knight, K. 2009. Fixed vs. Fluid vs. Elastic Layout: What's The Righ One For You? [viitattu 15.3.2015]. Saatavissa: http://www.smashingmagazine.com/2009/06/02/fixed-vs-fluid-vs-elastic-layout-whats-the-right-one-for-you/

Knight, K. 2011. Responsive Web Design: What It Is and How To Use It [viitattu 6.4.2015]. Saatavissa: http://www.smashingmagazine.com/2011/01/12/guidelines-for-responsive-web-design/

Korpela, J. 2009. JavaScript (ja vastaavat) [viitattu 7.3.2015]. Saatavissa: http://www.cs.tut.fi/~jkorpela/webjulk/3.2.html

Korpela, J. 2012. Responsiivinen suunnittelu [viitattu 29.3.2015]. Saatavissa: http://html5kirja.fi/2012/08/02/responsiivinen-suunnittelu/

Leiniö, T. 2012. Mitä on responsiivinen design? [viitattu 12.3.2015]. Saatavissa: https://www.sofokus.com/blogi/mita-on-responsiivinen-design/

LePage, P. 2014a. How to choose breakpoints [viitattu 13.3.2015]. Saatavissa: https://developers.google.com/web/fundamentals/layouts/rwd-fundamentals/how-to-choose-breakpoints?hl=en

LePage, P. 2014b. Use CSS media queries for responsiveness [viitattu 13.3.2015]. Saatavissa: https://developers.google.com/web/fundamentals/layouts/rwd-fundamentals/use-media-queries

LePage, P. 2015. Responsive Web Design Basics [viitattu 12.3.2015]. Saatavissa: https://developers.google.com/web/fundamentals/layouts/rwd-fundamentals/

Lerssi, P. 2014. Millaiset ovat hyvä verkkosivut? [viitattu 11.3.2015]. Saatavissa: paja.fi/millaiset-ovat-hyvat-verkkosivut/

Mening, R. 2013. Wordpress vs Jooma vs Drupal + CMS ”Comparison chart” [viitattu 3.4.2015]. Saatavissa: http://websitesetup.org/cms-comparison-wordpress-vs-joomla-drupal/

Mobiilimarkkinointi Routa Oy. 2015. Asiakkaasi tekevät ostopäätöksiä mobiilissa. Onko yrityksellläsi jo mobiilisivut? [viitattu 11.3.2015]. Saatavissa: http://www.mobiilimarkkinointirouta.fi/mobiilisivut

Nokia. 2015. Nokian nettisivu [viitattu 23.3.2015]. Saatavissa: http://company.nokia.com/fi

QR8 Web Design Blow. 2014. Website Design 101 [viitattu 9.3.2015]. Saatavissa: http://www.qr8design.com/2014/06/16/website-design-101

Quinta Group. 2015. Responsive web design – adapt, respond and overcome [viitattu 12.3.2015]. Saatavissa: http://quintagroup.com/services/web-design/responsive-web-design

Pure. 2014a. Buttons – Simple CSS for buttons. [viitattu 16.3.2015]. Saatavissa: http://purecss.io/buttons/

Pure. 2014b. Pure.css – A set of small, responsive CSS modules that you can yse in every web project. [viitattu 16.3.2015]. Saatavissa: http://purecss.io/

Raggett, D. 1998. Raggett on HTML4. Boston: Addison Wesley. [viitattu 7.3.2015]. Saatavissa: http://www.w3.org/People/Raggett/book4/ch02.html

Riippi, J. 2013. Natiivi, hybridi ja html5 [viitattu 12.3.2015]. Saatavissa: https://67prosenttia.wordpress.com/2013/05/27/natiivi-hybridi-ja-html5/

Roydee. 2010. Mastering the 960 Grid System [viitattu 5.4.2015]. Saatavissa: http://code.tutsplus.com/tutorials/mastering-the-960-grid-system--net-13794

Saarikumpu, O. 2014. Johdanto CSS-tyyliehdotuksien käyttöön Web-sivuilla [viitattu 7.3.2015]. Saatavissa: http://weppipakki.com/css/tekstit/cssintro.htm

Smith, N. 2015a. 960 GRID SYSTEM [viitattu 9.3.2015]. Saatavissa: http://960.gs

Smith, N. 2015b. CSS Documentation [viitattu 16.3.2015]. Saatavissa: http://unsemantic.com/css-documentation

Smith, N. 2015c. Desktop Grid [viitattu 16.3.2015]. Saatavissa: http://unsemantic.com/demo-responsive

Smith, N. 2015d. Unsemantic [viitattu 16.3.2015]. Saattavissa: http://unsemantic.com/

Ubinet. 2015. Julkaisujärjestelmät [viitattu 2.4.2015]. Saatavissa: http://www.ubinet.fi/www-suunnittelu/julkaisuj%C3%A4rjestelm%C3%A4

van Hove, N. 2015. CSS Media Queries [viitattu 13.3.2015]. Saatavissa: http://cssmediaqueries.com/

Verkkotaikurit. 2015. Nettisvujen suunnittelu, tekeminen ja julkaisu sekä muita ohjeita [viitattu 23.3.2015]. Saatavissa: http://www.webhotellivertailu2.fi/nettisivujen-suunnittelu-tekeminen-ja-julkaisu-seka-muita-ohjeita/

Virtanen, R. 2015. Websivun suunnittelu [viitattu 23.3.2015]. Saatavissa: http://www2.kyamk.fi/~zrivi/websuunnittelu/web1/03suunnittelu.html

W3C. 2012. A Short History of JavaScript [viitattu 7.3.2015]. Saatavissa: https://www.w3.org/community/webed/wiki/A_Short_History_of_JavaScript

W3C. 2014. HTML5 Differences from HTML4 [viitattu 7.3.2015]. Saatavissa: http://www.w3.org/TR/2014/NOTE-html5-diff-20141209/

W3C. 2015. HTML, The Web’s Core Language [viitattu 6.3.2015]. Saatavissa: http://www.w3.org/html/

Way, J. 2009. A Detailed Look at the 960 CSS Framework [viitattu 10.3.2015]. Saatavissa: http://code.tutsplus.com/articles/a-detailed-look-at-the-960-css-framework--net-2567

Weil, A. 2014. Don’t Let Responsive Web Design Cut Into Customer Satisfaction [viitattu 15.3.2015]. Saatavissa: http://www.yottaa.com/blog/mobile/dont-let-responsive-web-design-cut-into-customer-satisfa

WMHost. 2015. DevNet eCMS julkaisujärjestelmä [viitattu 3.4.2015]. Saatavissa: http://www.wmhost.com/palvelut/sovellukset/julkaisujarjestelma

WordPress. 2015a. About Us. [viitatu 3.4.2015]. Saatavissa: https://fi.wordpress.org

WordPress. 2015b. WordPress Suomen verkkosivut [viitattu 3.4.2015]. Saatavissa: https://fi.wordpress.org/