

Verkkosivun responsiivisuus ja sen kehittäminen - Case: LAURA™-palvelu

Tynkkynen, Nino

2017 Laurea

Laurea-ammattikorkeakoulu

Verkkosivun responsiivisuus ja sen kehittäminen
Case: LAURA™-palvelu

Nino Tynkkynen
Tietojenkäsittelyn koulutusohjelma
Opinnäytetyö
Huhtikuu, 2017

Nino Tynkkynen

Verkkosivun responsiivisuus ja sen kehittäminen - Case: LAURA™-palvelu

Vuosi 2017

Sivumäärä 32

Tutkimuksen tavoitteena oli selvittää mitkä ovat responsiivisen suunnittelun keskeisimmät elementit ja toteutustavat. Toimeksiantajayrityksenä toimi LAURA Rekrytointi Oy. Kehittämistehtävänä oli kehittää yrityksen rekrytointijärjestelmä LAURA™-palvelun responsiivisuutta hyödyntäen tutkimuksen tuomaa tietoa.

Opinnäytetyössä käydään läpi keskeisimpiä responsiivisessa suunnittelussa käytettäviä HTML-elementtejä ja CSS-tyylimäärityksiä sekä verkkosivujen laitekohtaisen mukautuvuuden kehittämiseksi käytettäviä toteutustapoja. Opinnäytetyö toteutettiin laadullisena tutkimuksena. Tutkimuksessa tutkimustapana käytettiin tapaustutkimusta ja sisällön analysointiin hyödynnettiin tutkimusmenetelmänä sisällönanalyysiä.

Tutkimuksen lopputulemana oli, että responsiivisuus ja sen suunnittelu voidaan toteuttaa usealla tavalla, mutta on osattava punnita käyttötarkoitukseen sopivin ratkaisu. Tutkielman tuotoksena valmistui toimeksiantajayrityksen hyödynnettäväksi tarkoitettu suunnitelma. Suunnitelmassa on kuvattu, kuinka responsiivista suunnittelua voidaan kehittää LAURA™-palvelun työnhakijoille tarkoitetuilla verkkosivuilla.

Asiasanat: responsiivinen suunnittelu, web-kehitys, HTML, CSS

Nino Tynkkynen

Development of Responsive Web Design on Webpage - Case: LAURA™ service

Year 2017

Pages

32

The purpose of this Bachelor's thesis was to research essential elements and different implementing methods of responsive web designing. The thesis was produced for LAURA Recruitment Ltd. The task was to improve the responsive web design on LAURA™ service by using this thesis' research results.

The thesis researches the fundamentals of HTML elements and CSS specifications and implementing methods of responsive web designing. The thesis is case study. Content analysis was the research method of the research.

The main result is that responsive web design can be engineered in many methods depending of clients and users' needs. The conclusion of this thesis was a responsive web designing implementing plan for the LAURA™ service. The implementing plan proposes how to improve the responsive web design on webpages which is meant for the applicants using the LAURA™ service.

Keywords: responsive web design, web design, HTML, CSS

Sisällys

1	Johdanto	6
2	Tutkimuksen lähtökohdat	6
2.1	Aiheen rajausta ja tutkimuskysymys	9
2.2	Toimeksiantajayritys ja LAURA™-palvelu	9
3	Keskeiset käsitteet	10
4	Responsiivinen suunnittelu	11
5	Responsiivisen suunnittelun elementit ja toteutustekniikat	12
5.1	Elementtien sijoittelu	12
5.2	Mediaehdot	13
5.3	Viewport	14
5.4	Taulukot	15
5.5	Teksti	16
5.6	Media ja kuvat	17
5.7	CSS Framework ja Grid System	18
5.8	JavaScript	18
6	Tutkimustapa ja tutkimusmenetelmä	19
6.1	Tutkimusaineiston analysointi	20
6.2	Validiteetti ja reliabiliteetti	20
7	LAURA™-palveluun sopivan toteutustavan valinta	20
8	Responsiivisuuden implementoinnin suunnitelma LAURA™-palveluun	24
9	Yhteenveto ja johtopäätökset	27
	Lähteet	29
	Kuviot	32

1 Johdanto

Käytämme nykypäivänä internetin tarjoamia palveluita päivittäin erikokoisilta laitteilta. PC-tietokone ei ole enää ainoa väline internetin selaamiseen ja monelle esimerkiksi matkapuhelin on tärkeä väline palveluiden käytön kannalta. Verkossa toimivien palveluiden onkin tällöin kyettävä palvelemaan meitä käytössä olevasta laitteesta riippumatta. Näistä käytetyistä laitteista merkittävin kasvu on tapahtunut juuri mobiililaitteiden käytössä. Viimeisimmän Suomessa tehdyssä tilastossa internetin käytöstä mobiililaitteilla selviää, että Suomessa 69 prosenttia 16-89-vuotiaista omistaa älypuhelimien ja näistä 90 prosenttia käyttää sillä internetiä. (Tilastokeskus 2015.) Tästä onkin kehittynyt verkkopalveluiden kehityksen osalta trendi. Sujuva ja vaivaton käyttäminen on olennaista, jotta kyseessä olevan palvelun kaikki potentiaali tulee esille. Mikäli sivusto ei pysty vastaamaan asiakkaiden tarpeisiin, se voi menettää tärkeitä käyttäjiä. Sitä vastoin sujuvasti ja miellyttävästi toimiva palvelu houkuttelee käyttäjiään käyttämään sitä kaikenlaisilla laitteilla.

Opinnäytetyössä selvitettiin millaisilla tavoilla on mahdollista kehittää jo olemassa olevan järjestelmän responsiivisuutta, eli tämän mukautuvuutta eri kokoisille laitteille.

Opinnäytetyön keskeinen tehtävä oli pureutua eri responsiivisuuden toteutustapoihin, jonka pohjalta toteutettiin suunnitelma toimeksiantajayritykselle sopivasta responsiivisuuden toteutustavasta.

2 Tutkimuksen lähtökohdat

Mobiililaitteen käyttäminen verkkosivujen selaamiseen on lisääntynyt kasvavissa määrin työnhaussa (Ziemann 2017). Mobiililaitteilla LAURATM-palvelun kautta etsii työpaikkaa noin yksi kolmasosa työnhakijoista. Suurin osa käyttäjistä tulee edelleen PC-tietokoneilta, eli kannettavilla ja pöytätietokoneilla. Myös työhakemukset jätetään yleensä PC-laitteilta. Mobiililaitteet toimivatkin avoimien työpaikkojen selaamiseen, mutta käyttäjät jättävät usein kuitenkin hakemuksensa kotikoneelta. (Miettinen 2017.) Mobiililaitteilta LAURATM-palvelun kautta työpaikkaa etsivien määrän kasvun myötä on kehitettävä verkkosivut palvelemaan käyttäjiä paremmin matkapuhelimilla.

LAURATM-palvelun onkin kehityttävä vastaamaan näitä tarpeita entistä paremmin ollakseen toimiva ja helppokäyttöinen palvelu laitteeseen katsomatta. Responsiivisuuden huomioiminen LAURATM-palvelussa kuuluu osaksi yleistä järjestelmän jatkuvaa kehitystyötä. LAURA Rekrytointi Oy kuuntelee jatkuvasti LAURATM-asiakkaiden palautteita ja tarve kehittää järjestelmän responsiivisuutta on ilmennyt saatujen asiakaspalautteiden myötä. Asiakaspalaute on koskenut lähtökohtaisesti työnhakijoita palvelevia sivuja. Saatu asiakaspalaute vaikutti suuresti kyseisen kehitystyön valintaan.

Tutkija aloitti työskentelyn LAURA Rekrytointi Oy:ssa korkeakouluharjoittelun kautta, jonka jälkeen on jäänyt töihin yritykseen. Tutkimuksen aiheen valintaan on siis vaikuttanut paitsi tutkijan kokemukset toimeksiantajayrityksen palveluksessa työskentelystä, myös sen kautta herännyt kiinnostus rekrytointialaan ja sen kehittämiseen. Tästä on hyötyä kehitystyön suunnittelussa, koska työn kautta empiirisesti omaksuttua informaatiota voidaan näin ollen hyödyntää tutkimuksessa kirjallisuuden ohella.

Nykyisin LAURATM-järjestelmän työnhakijoille näkyvä puoli mukautuu näytölle tämän resoluutiotarkkuuden mukaan. Työnhakijoille näkyvän LAURATM-palvelun lomaketta voidaan pitää keskeisenä osana responsiivista sisältöä, sillä se sisältää kaiken kyseisen verkkosivun tiedollisen sisällön. Tämän leveys on 770 pikseliä, jonka jälkeen pienempi resoluutioisilla näytöillä tämän pitäisi lähteä pienentymään näytön koon mukaisesti laitteesta riippuen. Responsiivisuus kuitenkin toteutuu verkkoselaimen selainikkunaa pienennettäessä. Tämä ei kuitenkaan enää nykyisin riitä, sillä näyttöjen resoluutio on kehittynyt huomattavasti viimevuosien aikana.

Kuviossa 1 on esitetty LAURATM-palvelun demoversio hakijoille tarkoitettu avoimien työpaikkojen listaussivu matkapuhelimella. Kuviosta näkee, kuinka kyseinen verkkosivu jää pieneksi, jolloin teksti ei ole ensisilmäyksellä luettavuudelta selkeää. Reunoille jää paljon tyhjää tilaa ja taulukkomuotoinen esitystapa tekee sivusta leveän. Kuitenkin tekstiä voidaan lähteä suurentamaan lähentämällä lähemmäksi sivun sisältöä, mutta tällöin taulukon rivi näkyy vain osittain kerrallaan. Lisäksi painikkeet ovat hyvin pieniä ja näitä voi olla mobiililaitteen kautta sormella haasteellista painaa. (Kuvio 1.)

LAURA™
elämäsi paras rekrytointi

Haluatko päivittää hakemustasi?
[Kirjaudu sisään.](#)
Unohtitko salasanasi? [Näe uusi tilisi.](#)

Hakukriteerit

Toimiala Kaupunki

Jokin näistä sanoista

Väh. 3 merkkiä per sana

[Haku](#) [Tytyrää](#)

Avoimia työpaikkoja: 11

Tehtävän nimi	Hakuaika päättyy	Kaupunki	Toimiala
Sijaisia lyhytaikaisin sijaisuuksin			Pääkaupunkiseutu
Merikapteeni	02.05.2017 12:00	Logistiikka	Pääkaupunkiseutu
Taistelukäynninvalvonta	25.08.2017 16:00		
Rakennusinsinööri	30.01.2018 13:00		Pääkaupunkiseutu
PHP-ohjelmoija	31.01.2018 15:00	IT	Helsinki
Asistentti	01.03.2018 16:00		
Rahastusasiainkuntaja	28.04.2018 16:00	Myynti	Helsinki
Huoltomies lentosemalä	01.05.2018 16:00		Tampere
Myyntikonsultti	11.05.2018 12:00		
Toimistopäällikkö	30.06.2018 23:59	Hallinto	Pääkaupunkiseutu
Testaaja	25.10.2018 10:00		

Mikäli et löytänyt itseäsi kiinnostavaa avointa työpaikkaa, voit jättää avoimen hakemuksen. Olemme sinuun yhteydessä, mikäli osaamistasi vastaava työpaikka tulee hakuun.

[Avoin hakemus](#)

[Työpaikkavaiht](#)

LAURA™
Rekisteriseloste

Kuvio 1: LAURA™:n demoversion hakijasisivut mobiililaitteelta katsottuna.

Nykyisin älypuhelimien resoluutiot ovat parhaimmillaan 4K-tasoa, jolloin näytössä on $3\,840 \times 2\,160$ pikseliä (Kharpal 2017). Vanhempiin älypuhelimiin verrattuna resoluutiot ovat lisääntyneet huomasti, jotka saattoivat olla alkujaan QVGA-resoluutioilla 240×320 pikseliä (Video4Change). Resoluutiopohjainen responsiivisuus ei tällöin enää palvele nykypäivänä käytössä olevia laitteistoja ja responsiivisuus täytyy kehittää jotain muuta toteutustekniikkaa hyödyntäen. Verkkosivuston laitekohtaiseen mukautuvuuteen vaikuttaa monet muutkin asiat, kuten esimerkiksi laitteen leveyden mitta pikseleinä ei aina vastaa CSS:än käyttämiä pikseleitä. Tämä johtuu siitä, että CSS-pikseleihin vaikuttaa myös käytettävän laitteen pikseleiden tiheys. Esimerkiksi iPhone 5-puhelimen 4 tuumaisen näytön resoluutio on 640×1136 pikseliä, jonka pikselitiheys on 326 ppi. Ppi-arvo kuvaa pikseleiden määrän yhdessä

tuumassa. Näistä muodostuu CSS:n pikselit, jotka ovat iPhone 5-puhelimella 320 x 568 pikseliä. Tästä nähdään, että verkkosivut ei siis käyttäydy ainoastaan näytössä olevien pikseleiden mukaan. (Stephen 2014.)

Jokaisella LAURA™-asiakkaalla on yksilöllinen hakijasivu, joko omalla yrityslugolla tai laajemmin yritysilmeellä. Koska LAURA™-palvelulla on noin 150 asiakasta, joilla kaikilla saattaa olla eroavaisuuksia sisällöllisesti sekä ulkoasulta, saattaa järjestelmään tehdyt muutokset näkyä asiakkaiden sivuilla eri tavoin. Kun järjestelmään tehdään kaikille yhtenäisiä muutoksia olisi kuitenkin toivottavaa, että muutokset näkyisi asiakaskohtaisesti toivotunlaisesti. LAURA™-palvelussa on käytössä järjestelmän CSS-tyylitiedosto, joka sisältää kaikkien asiakkaiden yhteiset tyylimäärytykset. Näiden lisäksi asiakkailla on käytössä järjestelmäkohtaisia CSS-tyylimäärytykset, joilla voidaan ylikirjoittaa yhteisesti asetettuja tyylimääritteitä.

2.1 Aiheen rajausta ja tutkimuskysymys

Opinnäytetyö on rajattu käsittelemään responsiivisuutta verkkosivuilla. Responsiivisuuden toteutustapojen selvitystyötä LAURA™-palveluun hyödynnetään koko järjestelmän laajuudelta työn toteutukseen, mutta tämän suunnitelman keskiössä on kuitenkin työnhakijapuoli. LAURA™-palvelun laitekohtaisen mukautuvuuden kannalta tärkein kehityskohde on mobiililaitteiden tuki työnhakijoille.

Lisäksi työssä sivutaan mobiililaitteen käytettävyyden näkökulmaa ja kuinka tämä tulisi huomioida verkkosivun responsiivisuudessa yleisellä tasolla. Tämä on tärkeää kokonaisuuden hahmottamisen kannalta. Opinnäytetyössä haetaan vastausta tutkimuskysymykseen: Mitkä ovat verkkosivun responsiivisuuden kehitystavat ja työvälineet sekä kuinka näitä hyödyntämällä voidaan kehittää verkkosivu laitekohtaisesti mahdollisimman mukautuvaksi?

2.2 Toimeksiantajayritys ja LAURA™-palvelu

Opinnäytetyön toimeksiantajana toimi LAURA Rekrytointi Oy. Yritys on aiemmin tunnettu nimellä Uranus Oy, joka vaihtui maaliskuussa vuonna 2017 LAURA Rekrytointi Oy:ksi. LAURA Rekrytointi Oy on ura- ja rekrytointialan yritys, joka on perustettu Uranus Oy:nä vuonna 1998 ja on Arena Partners Oy:n tytäryhtiö. LAURA Rekrytointi Oy:n rekrytointiportaali Uranus.fi palveluihin kuuluu työpaikka-, koulutus- ja yrittäjyystarjonta, jotka pyritään tarjoamaan työnhakijoille mahdollisimman helposti löydettävästi sekä haettavasti. Uranus.fi pyrkii huomioimaan mahdollisimman hyvin eri uratilanteissa olevia ihmisiä. Se tarjoaa palveluja esimerkiksi opiskelijoille ura- ja rekrytointipalveluja sekä vinkkejä piilotyöpaikoista työnhakijoille. Uranus.fi julkaisee työnhakuun liittyviä artikkeleita, joissa pyritään auttamaan

työnhakijaa työnhaun eri tilanteissa. Yhtiö on jäsenenä kansainvälisessä rekrytoinnin yhteistyöverkostossa The Networkissa. (Uranus.fi).

Nimenvaihdoksen lisäksi myös nykyinen Uranus.fi uudistuu ja tämän tilalle julkaistaan uusi Rekrytointi.com-uraportaali. Lanseeraus on tarkoitus tapahtua kevään aikana vuonna 2017, jonka myötä palvelu on muun muassa entistä parempi käytettävyydeltä ja siten myös helppokäyttöisempi. Rekrytointi.com-uraportaali tulee yhdistämään entistä vahvemmin LAURA Rekrytointi Oy:n päätuotteita yhteen, jolloin uraportaalin ja LAURA™-palvelun tarjoamat palvelut ovat entistä lähempänä toisiaan. (Uranus Oy 2017.)

LAURA™-palvelu on kattava rekrytoinnin hallintaa edistävä järjestelmä, joka helpottaa rekrytoinnin aikana muun muassa rekrytoinnin suunnittelua, työpaikkojen hallintaa, hakemuksien käsittelyä ja oikeiden henkilöiden valintaa. Olettamuksena on, että yritykset haluavat tehdä rekrytointiprosessinsa helpoiksi ja tehokkaiksi. Tähän tarpeeseen LAURA™-palvelu pyrkii vastaamaan. Oikean työntekijän valinnan jälkeen järjestelmässä on mahdollista hallita erilaisia henkilöstöön liittyviä asioita kuten työsuhdetta, sijaisuuksia ja sisäisiä rekrytointeja. Myös yhteistoimintamenettelyä prosesseineen on mahdollista hallita järjestelmällä. Lisäksi viestintä hakijoiden kanssa ja raportointi rekrytoinnista onnistuvat järjestelmän kautta helposti. (LAURA). LAURA™-palvelu on käytössä tällä hetkellä noin 150 asiakkaalla. Järjestelmän kautta kulki viime vuonna yli 420 000 työhakemusta (Miettinen 2017.) Palveluita on erilaisina versioina, jotka ovat LAURA™ Solo, LAURA™ Start ja LAURA™ Pro. Näistä Solo on suunniteltu yksittäisen rekrytoivan henkilön käyttöön. Start ja Pro ovat laajempia palveluiltaan ja näistä Pro räätälöidään aina asiakastarpeiden mukaisesti sopivaksi. (LAURA.)

3 Keskeiset käsitteet

Responsiivinen suunnittelu on keskeisin käsite opinnäytetyössä. Responsiivinen suunnittelulla viitataan englanninkieliseen termiin *Responsive web design*. Responsiivisuus ilmenee verkkosivuston sisällön mukautuvuutena käytettävän päätelaitteen näytön koon mukaiseksi. Responsiivinen suunnittelu käsitteenä kattaa kaikki verkkoa käyttävät laitteet. (W3Schools, a.) Responsiivisen suunnittelun käsitteen luojana pidetään Ethan Marcottea, joka kirjoitti artikkelin aiheesta vuonna 2010. (Kadlec 2013, 11).

Mobile First Design käsitettä käytetään myös opinnäytetyössä. Käsitteenä tämä painottaa responsiivisuutta mobiililaitelähtöisesti, jolloin verkkosivun sulautuvuutta suunnitellaan ensisijaisesti mobiililaitteille ja toissijaisesti muille laitteille (W3Schools, b).

HTML eli *the Hypertext Markup Language* on yhtenäinen kieli verkkosivustoille. Verkkosivu on verkkoselaimelle luettavaksi tehty HTML-dokumentti. Opinnäytetyössä on valittu verkkosivu

keskeiseksi käsitteeksi, sillä se on terminä yhtenäinen kaikille verkossa näkyville sivustoille. (W3 2014.) Verkkosivu on käsitteenä tällöin yhtenäinen responsiivisuuden kanssa, sillä responsiivinen suunnittelu käsitteenä liitetään juuri verkkosivuihin.

CSS eli *Cascading Style Sheets* on kieli, joka määrittää verkkosivun esitysmuodon ja ulkoasun. CSS-kielillä voidaan määrittää muun muassa, että mitä fontteja ja värejä sivustolla käytetään. (W3 2014.) Esimerkiksi verkkosivustoilla esitettävien elementtien sijaintia, kokoa ja muotoa voidaan muokata esitysmuodolta mieluisaksi CSS-kielillä.

4 Responsiivinen suunnittelu

Erikokoisia verkkoselausta hyödyntäviä laitteita julkaistaan kasvavissa määrin. Verkkoselaaminen on mahdollista nykyisin pöytätietokoneen lisäksi muun muassa älytelevisiolla, pelikonsolin tai matkapuhelimen avulla. Laite voi pienimmillään olla nykyisin jopa älykello (Griffin 2015).

Jotta kaikki laitteet näyttäisivät verkkosivun sisällön halutunlaisesti, tarvitsemme yhtenäisen tekniikan tämän toteuttamiseen. Responsiivisella suunnittelulla pyritäänkin saamaan verkkosivustot näyttämään sivut mahdollisimman selkeästi laitteen ominaisuuksista riippumatta. (Kadlec 2013, 4-6.) Hyvin toteutetulla responsiivisella suunnittelulla sivusto saadaan näkymään laitekohtaisesti luettavasti ja selkeästi (Kuvio 2).



Kuvio 2: Responsiivinen verkkosivu kuvattuna erilaisilla laitteilla.

Responsiivinen suunnittelu sisältää joukon erilaisia tekniikoita, joilla laitekohtaista mukautuvuutta voidaan toteuttaa. Ethan Marcotten (2010) kirjoittama artikkeli on ensimmäinen artikkeli, jossa mainitaan responsiivisesta suunnittelusta. Hän mainitsee

artikkelissa kolme teknistä vaatimusta responsiiviselle suunnittelulle: joustavat ruudukot eli *fluid grids*, joustavat kuvat sekä CSS-mediaehdot (Marcotte 2010).

5 Responsiivisen suunnittelun elementit ja toteutustekniikat

Suunnitellessa sivustoja responsiiviseksi lähtökohta on huomioida erilaisten ja erikokoisten laitteiden laaja-alaisuus. Verkkosivujen responsiivisuutta voidaan lähteä toteuttamaan eri toteutustavoin. Taustalla voi olla esimerkiksi erilaiset käyttäjätarpeet tai sivuston käyttötarkoitus. Sivuston mukautuvuutta kannattaakin lähteä kehittämään näiden tarpeiden pohjalta. Responsiivisuuden toteutus voi olla tarpeiden pohjalta laitekohtaisesti toteutettua. Toinen tapa on toteuttaa mobiililaitelähtöisesti. (Marcotte 2011.)

Toteutus voidaan tehdä lähtökohtana toteuttaa verkkosivu kaikille laitteille sopivaksi, jolloin voidaan ajatella sisällön käyttäytyvän ikään kuin neste. Neste on tässä toteutustekniikassa hyvä esimerkki, sillä neste mukautuu kohde objektin mukaiseksi luonnostaan. Samoin verkkosivusto voidaan ajatella mukautuvan tätä selailevan laitteen näytölle omassa niin sanotussa luontaisessa muodossaan. Tämä viimeisin tapa on responsiivisen suunnittelun kannalta suositeltavin, sillä lisääntyneiden laitemäärien myötä ei voida tietää millä laitteilla ihmiset selaavat verkkosivuja. (Marcotte 2011.)

5.1 Elementtien sijoittelu

Responsiivisuuden suunnittelun keskeinen osa on oikeanlaisten elementtien sijoittelumallien valinta. Layout voidaan toteuttaa usealla eri tavalla, mutta tässä kappaleessa keskitytään selvittämään responsiivisuuden kannalta keskeisimpiä tekniikoita. Responsiiviset layout tekniikat perustuvat mittasuhteiltaan prosenttilukuihin ja niitä kutsutaan termillä *Fluid layouts*. Perinteisesti CSS-kieltä käytettäessä mittasuhteet on määritelty niiden pikseliarvojen mukaan, joka kertoo tämän tarkan mittasuhteen. Prosentuaalisesti määritetty elementin mittasuhte joustaa annetun containerin leveyden ja korkeuden mukaan. Containerin tarkoituksena on kehystää elementtinä tämän sisälle halutut muut verkkosivustoilla käytettävät elementit (Kadlec 2013, 21-25.)

Toteutustavoille yhtenäistä on ajattelumalli, jossa verkkosivun elementit sijoitellaan ikään kuin ruudukkoon. HTML-elementtien joustava sijoittelu ruudukkoihin voidaan toteuttaa monella eri tavalla. Elementit asetellaan ruudukon pystyriveihin ja vaakariveihin riippuen toteutustavasta. Keskeistä joustavassa ruudukossa on, että elementtien leveydet on toteutettu prosentuaalisilla arvoilla. Lisäksi *box-sizing* määritetään käyttämään *border-box* nimistä arvoa kaikille elementeille, jolloin elementeissä käytetty täyte eli *padding* sisältyy näille määritettyihin mittoihin. Elementtien sijoitteluun voidaan hyödyntää erilaisia CSS-

tyylimäärittäjiä. Toteutus voidaan tehdä esimerkiksi kellutuksella eli floatingilla. (w3schools, c.)

Uusia tekniikoita kehitetään jatkuvasti lisää. Yksi elementtien sijoittelutavoista on *grid layout* suomeksi hila, jonka avulla voidaan jakaa verkkosivu määritettyihin osiin eli ruudukkoihin. Menetelmällä sivuille voidaan sijoitella elementit ruutuihin ja valita näistä kuinka monta ruutupaikkaa elementti näistä käyttää. (Korpela 2013 94-95). Tämä tekniikka mahdollistaa muun muassa elementtien nimeämisen CSS-tyylimäärittäyksissä, joka mahdollistaa HTML-elementtien uudelleen sijoittelun suoraan CSS-määrittäyksistä, joka voidaan tehdä CSS grid layoutissa käyttämällä nimettyjä ruudukko rivejä. (W3 2017.) Haluttu sivustorakenne lisätään grid-template-areas-määrittäykseen (Kuvio 3).



Kuvio 3: Esimerkki nimetyistä ruudukko riveistä, vasemmalla CSS grid layout määrittäykset ja oikealla lopputulos.

Kuitenkaan CSS Grid layout ei ole kovin tuettu selaimissa. Tuki on lisääntynyt 2017 vuoden alussa suurimmassa osassa selaimista ja tunnetuimmat selaimet tukevatkin jo kyseistä tekniikkaa. (Can I use, 2017a.) Kuitenkin vanhempien selainversioiden käyttö sekä muut tätä tukemattomat selaimet huomioiden on hila ja muissa ei-tuetuissa tekniikoissa syytä tehdä varmistavia rakenteita. Jos selain ei tue tekniikkaa, se palaa elementin oletusasetuksiin. Puutteisiin voidaan hyödyntää esimerkiksi The Modernizr JavaScript library nimistä työkalua, joka testaa eri selaimien tukea halutulla verkkosivulla. (Hampton-Smith 2016, 127.)

Flex boxes eli joustolaatikot ovat myös sopivia responsiivisen suunnittelun mukaiseen elementtien sijoitteluun. Näiden hyöty on se, että verrattuna float-tyyppisiin elementteihin nämä elementteinä sisältävät padding eli elementin reunuksien määrittäykset. Näitä voidaan lataa mieleisiin suuntiin. Elementit voidaan määrittää niin, että toinen elementeistä on esimerkiksi kolme kertaa toista elementtiä leveämpi. (Korpela 2013, 91-92).

5.2 Mediaehdot

Media query eli mediaehdot ovat laitekohtaisia asetuksia, joita määrittämällä saadaan verkkosivun sisältö näkymään laitekohtaisesti. Mediaehdoilla saadaan verkkosivut käyttäytymään halutulla tavalla kohdistamalla määrittäjiä laitteiden näyttöjen leveyksien

mukaisesti. Mediaehtoihin voidaan tehdä ehtoja kohdistetusti joillekin laitteille, kuten tulostamista varten määrittämällä mediaksi print. Mediaehto voi kohdistua myös useampiin medioihin käyttämällä media määritteenä all. Ehtoihin voidaan määrittää laitteen näytön leveys, jolloin voidaan asettaa kyseisille laitteille sopivat elementtien sijoittelut ja muut tyylisäädöt. (Korpela 2013 72-76.)

Mediaehdot voidaan toteuttaa sulautettuna tai ulkoistettuna pääasiallisesta tyylitiedostosta (Kuvio 4). Sulautettuna ne ovat samassa tiedostossa, jolloin HTTP-lataus sujuu nopeammin tai vaihtoehtoisesti voidaan ulkoistaa CSS-asetukset eri laitteille omiksi tiedostoiksi. (Kadlec 2013, 73-75.)

```
<!-- Ulkoinen lähde -->
<link rel='stylesheet' media='screen and (max-width: 960px)' href='css/esimerkki.css' />
<style>
/* Sisäinen lähde */
@media all and (max-width: 600px) {
/* Tyylimäärittelyt tänne */
}
</style>
</head>
<body>
```

Kuvio 4: CSS-mediaehdot sisäisessä ja ulkoisessa muodossa.

Mediaehdot voidaan lähteä rakentamaan PC-tietokoneesta pienempiin näyttöihin, jolloin kehitys aloitetaan suuri näyttöisestä pienempiin. Toinen vaihtoehto on lähteä kehittämään mobiilista isompiin laitteisiin. Yleisempää on kehittää sivustoja PC-tietokone lähtöisesti, jolloin sivuston CSS-määrittelyt ovat tehty kannettavien ja pöytätietokoneiden näyttöalueen mukaisia. Tämän jälkeen mediaehdoilla voidaan asettaa laitekohtaisesti eri ehtoja. Ehtoihin esimerkiksi voi kuulua, että lisätään linkkeihin ikonit mobiililaitteilla tai voidaan muuttaa elementtien sijoittelua. Kuitenkin aloittamalla kehittäminen mobiililähtöisesti, voidaan välttyä monilta ongelmilta, joita pöytätietokone lähtöisessä suunnittelussa ei huomata. (Kadlec 2013, 73-75.)

5.3 Viewport

Viewport vastaa selainikkunan leveyttä, jolla voidaan määrittää tähän liittyviä asetuksia. Se voidaan määrittää verkkosivun metatiedoissa tai suoraan CSS-määrittelyyn, jonka tuki on metatietoon verrattuna heikompi. Viewportin avulla voidaan määrittää verkkosivustolle, että kuinka verkkosivu näkyy laitteilla. Tämän avulla voidaan määrittää, kuinka paljon käytettävällä laitteella on mahdollista käyttää suurennusta ja loitonnusta alkuperäisestä koosta. (Kadlec 2013, 57-64.)

Viewportin metatietoihin voidaan määrittää tämän nimeksi viewport, joka kuvaa sen, että mitä tietoa meta-tag sisältää (w3schools, d.) Tämän content osan sisälle voidaan määrittää metatiedon sisältöä. Contenttiin voidaan syöttää width-arvoksi device-width, jolloin verkkosivut saadaan mukautumaan laitteen leveyden mukaan (Kuvio 5).

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Kuvio 5: Viewportin määrittäminen.

Määrittämällä initial-scale arvoon 1, sivusto säilyttää alkuperäisen etäisyyden ensimmäisestä sivun latauksesta. Näillä asetuksilla saataisiin toteutettua responsiiviseen suunnitteluun sopiva ratkaisu. Kun sivun sisältö on määritetty joustavana ruudukkona viewport osaa asettaa sivun sisällön tämän määrittämisien mukaisesti. (Kadlec 2013, 57-64.)

5.4 Taulukot

HTML-tilutukot eivät ole lähtökohtaisesti rakenteeltaan laitekohtaisesti mukautuvia, sillä taulukkorakenne on tehty esittämään tietoa leveyssuunnassa. Taulukot vaativat tällöin mobiililaitteilla erityishuomiota, sillä mobiililaitteet esittävät tietoa kapeammalta näytöltä kuin PC-tietokoneet. Jotta taulukot saataisiin toteutettua responsiivisen suunnittelun mukaisesti, voidaan taulukon sisältöä joutua asettelemaan erilaiseen esitysmuotoon.

Pieninäyttöisten laitteiden kohdalla kannattaa miettiä, että mikä taulukon sisällöstä on tarpeellista ja mitkä näistä tiedoista on tarpeen näkyä kaikilla laitteilla? Taulukon responsiivisuutta suunnitellessa on tärkeää tietää, että kuinka sivustolla vierailevat käyttävät taulukon tietoa. Myös on hyvä selvittää, että mihin asti taulukko on nykyisellään luettavaa. Taulukko voidaan esittää responsiivisen suunnittelun keinoin tyypillisesti kolmella eri tavalla: uudelleen määrittämällä taulukon sarakkeiden koot, uudelleenasettelemalla taulukko tai rajaamalla taulukon tiedoista näytettävää sisältöä. (Kyrnin 2015, 225-228.) Näiden lisäksi yksi yleinen usein verkossa näkyvä toteutustapa on vieritettävä taulukko, jossa taulukko esitetään pienemmillä näytöillä vierityksen avulla. Vieritystä käyttäessä taulukko voi pienentyä pienimpään luettavuuden kannalta olevaan tilaan. (Rizzo 2016). Nopein toteutustapa on sarakkeiden kokojen uudelleen määrittäminen, jolloin taulukon leveydeksi määritetään 100%. Kun taulukko on määritetty prosentuaalisesti, se mukautuu selainikkunan leveyteen. Kuitenkin näyttöalueen ollessa pieni voi teksti mennä luettavuudelta epäselväksi. (Kyrnin 2015, 227-229.) Tämä voidaan kuitenkin estää, kun tämän jälkeen lisätään vieritysmahdollisuus ennen luettavuuden heikkenemistä.

Uudelleenasetellessa voidaan purkaa taulukko näyttämään ikään kuin jokainen taulukon tietorivi olisi oma elementtinsä. Näin silmälaitteille saadaan mobiililaitteille helpommaksi, sillä tieto on esitettyä omana elementtinä, jossa saman taulukon rivin tieto on esitetty allekkain. Mobiililaitteella ei tällöin tarvitse vierittää verkkosivua vaakasuunnassa. Purkamiseen voidaan hyödyntää monia eri keinoja, mutta tämän rakentaminen on mahdollista myös puhtailla CSS-määrittelyillä. (Wallace 2015.)

5.5 Teksti

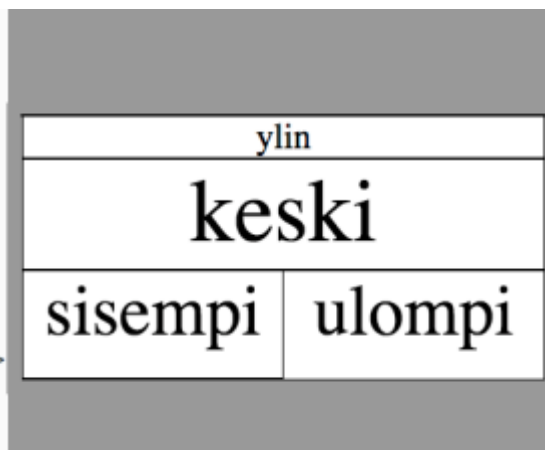
Verkkosivun tekstit tulee huomioida responsiivisuutta suunnitellessa. Teksti voi näyttää esimerkiksi suuremmalta älypuhelimella kuin PC-tietokoneella, jos sivuilla on käytössä ainoastaan pikselein määritetyt fonttiarvot ilman mediaehtoja. Liian pieni tai liioitellun suuri teksti voi olla käytettävyyttä heikentävä tekijä aivan kuin muutkin ominaisuudet. CSS3 tarjoaa kolme aikaisempiin versioihin verrattuna uutta arvoa vw, vh ja vmin fonttien koon määrittämiseen. 1 vw on yhtäkuin 1% viewportin leveydestä, kun taas 1 vh vastaa 1% viewportin korkeudesta. Vmin reagoi pienempään vw- tai vh-arvoon ja samoin vmax-arvo vertaa kumpi näistä on korkeampi ja käyttää tämän arvoa. Näitä arvoja käyttämällä teksti reagoi heti näytön koon muutoksiin esimerkiksi pienennettäessä selainikkunan kokoa. (Coyier 2012a.)

Toinen toimiva tapa on asettaa isäntä elementtiin määritetyn fontin koon arvoksi esimerkiksi sivuston body:n tyylimäärityksiin pikseleinä arvo, joka voisi olla 16 pikseliä. Käytettävä yksikkö voi olla myös prosentuaalinen. Kun verrattava arvo on asetettu, voidaan ruveta käyttämään em arvoja. Nämä ovat skaalautuvia ja verrattavissa annettuun korkeammalle tasolle HTML:ssä asetetun isäntä elementtiin määritettyyn fontin koon pikseliarvoon. 1 em vastaisi esimerkkitapauksessa 16 pikseliä ja vastaavasti 1.5 em verrattuna 16 pikseliin tämän arvoksi muodostuisi 24 pikseliä. Vastaavasti voitaisiin verrata isäntä elementin fontin koon arvoa prosentuaalisella arvolla, jolloin 100% vastaisi pääfontin koon määrittelyn arvoa. Esimerkkitaapauksessa oli käytetty 16 pikselin arvoa, koska tämä on keskimääräisesti katsottuna selaimien yleisimmin käytössä oleva oletusfontti koko. (Kadlec 2013, 28-36.) Kuviossa 6 on kuvattu, kuinka em-yksikköä voidaan käyttää tekstissä (Kuvio 6).


```

30 body {font-size:16px}
31 #ylin {font-size: 1em} /*16px*/
32 #keski {font-size: 2.5em} /*40px*/
33 #ulompi {font-size: 2em} /*32px*/
34 .sisempi {font-size: 1em} /*16px*/
35 </style>
36 </head>
37 <body>
38   <div id="container">
39     <div id="ylin">ylin</div>
40     <div id="keski">keski</div>
41     <div id="ulompi">ulompi
42       <div class="sisempi toka">sisempi</div>
43     </div>
44   </div>
45 </body>

```



Kuvio 6: em-yksikkö käytettynä fonttien koko määrittämisessä.

Lisäksi fonttien kokoa määrittävistä yksiköistä *root em* eli rem, sopii responsiiviseen suunnitteluun. Tämä on toiminnoiltaan samalla tavoin skaalautuva kuin em, mutta skaalaus perustuu aina juurielementin tyylimäärittäisiin määritettyihin fonttien kokoihin, jotka toimivat sivuilla muille käytetyille fonteille skaalauksen vertailuarvona. Juuren fontti koon yksikkönä voidaan käyttää pikseleitä tai prosentteja samaan tapaan kuin em-yksikön kanssa. Myös skaalautuvuus toimii samalla tavalla kuin em-yksiköllä, eli juurielementin ollessa 16px niin 1 rem juurielementin alaisissa määrittämissä vastaa 16 pikseliä.

Rem korjaa kuitenkin muutamia em-yksikön ongelmia. Joissakin tapauksissa em-yksikkö ei osaa kohdistaa body tagin alla olevaa määrittystä isännäksi, sillä em-yksikkö perustuu isäntäelementteihin. Tämä näkyy listaelementtejä `` käytettäessä, jolloin em yksiköt hakevat listaelementin isäntäelementtiä oman määritetyn vertailuarvollisen arvon sijaan. Tähän verrattuna rem taas käyttää aina juureen määritettyä arvoa vertailuarvonaan, eikä vastaavia ongelmia rem-yksikköä käytettäessä ilmene. (Snook 2011.)

5.6 Media ja kuvat

Responsiivisessa suunnittelussa huomioitavia mediaelementtejä voivat verkkosivuilla olla erilaiset videot ja kuvat. Näiden sijoittelu tulee huomioida uudelleen, kun laitteen näytön koko on pieni. Videoiden ja kuvien tulee olla helposti katsottavissa ja turhaa sivun vieritystä on hyvä välttää. Käyttäjäkokemuksen kannalta pienellä näytöllä voidaan miettiä, että onko esimerkiksi artikkeliin ohjaavat kuvat välttämättömiä, vai riittääkö pelkästään artikkelin otsikko. Poistoon kannattaa hyödyntää JavaScriptiä, sillä pelkkä `display: none`-määrittäminen vain piilottaa kuvan ja lataa tämän normaalisti taustalla, jolloin verkkosivu latautuu hitaammin. (Kadlec 2013, 98-101.) Artikkelin mennessä voisi kuva näkyä, mutta tämänkin kohdalla voidaan tehostaa ruudun tilan käyttöä. Kuvaa voidaan näyttää eri versiossa pienille näytöille, jolloin tiedostokokoa voi olla esimerkiksi puolet alkuperäisestä. Kuvaa voidaan tällöin tehostaa leikkaamalla tätä ja näyttämällä ainoastaan keskeisin osa kuvaa. Responsiivisuuden lisäksi

muutokset nopeuttavat sivunlatautumista mobiililaitteilla, joilla lataus on muutenkin hitaampaa. (Kadlec 2013, 98-101.)

HTML5-ominaisuuksiin kuuluu `<video>` tunniste, jonka sisälle voidaan määritellä suoraan videon lähde. Elementtiin on nyt helppo tarttua CSS-kielen avulla, jolloin voidaan määritellä tälle helposti leveydeksi 100% ja korkeudeksi auto-arvo. Tämän myötä video reagoi näytön kokoon responsiivisesti. Myös kuvat voidaan samoilla leveys ja korkeus arvoilla tehdä näytölle mukautuvaksi. (Kyrnin 2015, 199-218.)

5.7 CSS Framework ja Grid System

CSS Framework on paketti erilaisia valmiita CSS:n asetuksia. Pakettien tarjoajia on useita ja nämä eroavat toisistaan ominaisuuksiltaan. Keskeisiä ominaisuuksia näille on valmiit grid system ratkaisut, jotka palstoittavat HTML-elementit ruudukkoihin. CSS Frameworkit voivat lisäksi sisältää monia erilaisia valmiita CSS-tyylittelyratkaisuja, kuten valmiiksi tyyliteltyjä pudotusvalikoita ja painikkeita. Tämä tuo muun muassa pudotusvalikoiden osalta helpotusta siihen, että eri selaimet käsittelevät nämä oletusarvoisesti eri tavoin ja tuottavat näille oman ulkoasun. Näiden ulkoasumäärittelyt ovat usein hyvin työläisiä ylikirjoittaa. (Coyier 2012b.)

Tällä hetkellä yksi suosituimmista ja laajimmista CSS frameworkeistä on Bootstrap. Se sisältää HTML, CSS ja JavaScript frameworkit. (Bootstrap.) CSS Frameworkit voivat kuitenkin alkaa koostumaan ominaisuuksien puolesta raskaiksi. Aina ei projektissa välttämättä ole kuitenkaan tarvetta kokonaiselle CSS Frameworkille. Useammat näistä CSS Frameworkkien tarjoajista mahdollistavat kuitenkin ominaisuuksien valitsemisen, jolloin voidaan valita juuri omaan tarpeeseen sopivia ominaisuuksia. (Smith 2013.)

Aina ei välttämättä tarvita kokonaista CSS Framework-ratkaisua, jolloin hyvä ratkaisu voi olla ottaa käyttöön ainoastaan Grid System. Valmiita Grid System-paketteja on monia, jotka on toteutettu erilaisilla ratkaisuilla. Useat CSS Frameworkit sekä Grid Systemit ovat toteutettu avoimena lähdekoodina, jolloin lähdekoodi on käyttäjän luettavissa ja ovat lähtökohtaisesti ilmaisia sekä vapaasti käytettävissä. (Gube 2013.)

5.8 JavaScript

JavaScriptin avulla voidaan tehostaa monia erilaisia responsiivisuuden suunnittelun osa-alueita. Responsiivisessa suunnittelussa voidaan hyödyntää Unobtrusive JavaScriptin keinoja, joka suomennettuna tarkoittaa huomaamatonta JavaScriptiä. Lähtökohtana huomaamattomassa JavaScriptissä on kehittää JavaScript erillään HTML-elementeistä, jottei käyttö rajoittuisi käyttäjillä, jotka eivät käytä JavaScriptiä. Verkkosivun rakenne kuvataan

tällöin CSS:ää ja HTML:ää käyttäen, jolloin JavaScriptillä lisätään toiminnallisuutta sivuille. (Lennon 2010.)

Jos verkkosivusto on rakennettu syöttäen JavaScriptiä esimerkiksi HTML-elementin inline-määrittelyyn, voi tulla ongelmia myös mobiililaitteilla. Joissakin tapauksissa voi olla hyödyllistä käyttää JavaScript pohjaisia ratkaisuja, joissa toiminnallisuus muutoin rajoittuisi esimerkiksi mobiililaitteen kosketusnäytön vuoksi. Painikkeisiin voidaan lisätä toiminnallisuuksia, jotka ilman JavaScriptiä eivät olisi tuettuina. Sormea pidettäessä painikkeen päällä, voidaan painikkeeseen luoda omat tyylittelyt :hover-selektorin tavoin. :hover on oma tyylimäritys HTML-painikkeille, kun hiiri laitetaan tämän painikkeen päälle. (LaGrone 2013, 161-163.)

Kuvien ja eri elementtien kohdalla voidaan määrittää mille laitteille sisältöä näytetään, kuitenkin hidastamatta sivulatausta. JavaScriptin natiivilla MatchMedialla voidaan valita näytettävät mediat ja muut elementit media queryn mukaisesti, jolloin mobiililaitteella voidaan näyttää rajatumpia pienempikokoisia kuvia sekä rajata tarvittaessa mobiililaitteella käytettävyyttä haittaavaa sisältöä. MatchMediaa käyttämällä verkkosivu lataa ainoastaan media queryn mukaisesti määritellyt asetukset, jolloin tämä ei lataa tarvitsematonta sisältöä. (Kadlec 2013, 98-101.) JavaScript toimii siis HTML- ja CSS-kielten lisänä ja tehostaa näiden toimintaa.

6 Tutkimustapa ja tutkimusmenetelmä

Opinnäytetyö toteutettiin laadullisena tutkimuksena. Tutkimustapana työssä käytettiin tapaustutkimusta. Kyseisen tutkimustavan valintaan vaikutti se, että tapaustutkimus soveltuu selkeästi rajatun aiheen tarkasteluun. Moni käsittää tapaustutkimuksen virheellisesti tutkimusmenetelmäksi, mutta sen sijaan se on käsitteenä monisyinen ja sitä voidaan siten soveltaa hyvin monenlaisiin tutkimuksiin ja kehitystöihin. Tapaustutkimus keskittyy keräämään yksittäisestä tapauksesta tai joukosta toisiinsa suhteessa olevista tapauksista yksityiskohtaista sekä intensiivistä tietoa. Käsiteltävän aineiston on olennaista muodostaa kokonaisuus, eli tapaus. (Eskola & Saarela-Kinnunen 2010, 190.)

Tutkimuksellinen kehittämistoiminta on keskeisessä osassa tätä kyseistä opinnäytetyötä. Tutkimuksellinen kehittämistoimintaa voidaan käyttää tuotettaessa arviointitietoa kehittämistoimien onnistumisesta tai vaikutuksista, tai tutkimustiedolla voidaan pyrkiä hyvien käytäntöjen konstruktointiin, tai mahdollisesti tutkimuksen avulla voidaan pyrkiä tukemaan kehittämistoimintaa. (Toikko & Rantanen 2009, 11-21.)

6.1 Tutkimusaineiston analysointi

Tutkimuksen kohteena on tutkimusaineiston sisällölliset ominaisuudet. Tutkimusmenetelmänä tutkimusaineiston tarkastelussa hyödynnetään sisällönanalyysiä, jolla etsitään aineiston merkityssuhteita ja merkityskokonaisuuksia. Tutkimusprosessissa sisällönanalyysi näkyi ennen kaikkea sisällön erittelynä. (Vilkkä 2005, 139-140.)

Sisällönanalyysin avulla työstä pyritään löytämään tutkimusaineistosta sopivimmat toteutustekniikat sekä tarkastelemaan tutkimuksessa käytetyn sisällön merkityksellisyyttä peilaten laajemmin opinnäytetyön tapaukseen. Tutkimusaineisto näin ollen on vertailevasti jaoteltu erilaisiin sarakkeisiin, joiden järjestystä on muuteltu analyysin aikana parhaimman toteutustavan löytämiseksi.

6.2 Validiteetti ja reliabiliteetti

Validiteetilla tarkoitetaan tutkimuksen luotettavuutta ja sen tarkastelu on tärkeä osa tutkimusprosessia (Metsämuuronen 2005, 109). Validiteetin avulla voidaan selvittää miten saatu tutkimustulos vastaa tutkittavaa todellisuutta (Heikkinen 2010, 153).

Tutkimuksen reliabiliteetti puolestaan tarkoittaa käytettävän menetelmän tai mittarin luotettavuutta ja käsitteellä arvioidaan käytettyjen tutkimusmenetelmien ja mittareiden kyvykkyyttä tuottaa näillä tavoiteltavia tuloksia. Reliabiliteettiä arvioidessa tulee ottaa myös huomioon tehtyjen havaintojen pysyvyys eri aikoina. (Saaranen-Kauppinen & Puusniekka.) Reliabiliteetti toimii välineenä, kun etsitään satunnaisia vaikuttavia tekijöitä tutkimusmenetelmästä (Heikkinen 2010, 153).

7 LAURA™-palveluun sopivan toteutustavan valinta

Nykyiset LAURA™-asiakkaiden rekrytointiin tarkoitettu työpaikkailmoitusten julkaisusivut, eli työnhakijoille näkyvät sivut on jo osittain kehitetty mukautumaan sivuilla vierailevien näyttölaitteille. Hakijasivuilla työpaikkailmoitukset ja työhakemuslomakkeet perustuvat lomakemuotoiseen tiedon esitystapaan. Sisältö on tällöin lähtökohtaisesti aseteltu selkeästi allekkain, eikä sisältöä ole suuremmin aseteltu vaakasuorassa vierekkäin. Nykyisin järjestelmässä ei ole kuitenkaan käytössä kuin muutamia mediaehtoja ja eikä ollenkaan viewport-määrittäjiä.

Kuitenkin sisältö reagoi lähtökohtaisesti melko hyvin jo nykyisellään selainikkunan pienentämiseen (Kuvio 7). Kun viewport ja metaehdot otetaan käyttöön sekä säädetään nämä mobiililaitteille sopiviksi, myös hakemuslomakkeen tekstikenttien tulisi mukautua jo lähtökohtaisesti melko hyvin. Kuitenkin mobiililaitteella voidaan optimoida sisältö mediaehtoien avulla hyödyntämään koko puhelimen näytön tila, jolloin lomakkeen sisältöä

voisi levittää koko näytön leveydelle. Mediaehdoissa voidaan poistaa kohdistettuna mobiililaitteiden leveyksille ylimääräistä täytettä eli *padding*-määrittäjiä, jotka luovat lomakkeen reunoilla esiintyvän tyhjän tilan.



Kuvio 7: Nykyinen hakemuslomake kavennetulla verkkoselaimen ikkunalla.

Elementtien sijoittelussa täytyy huomioida LAURA™-palvelua käyttävien asiakkaiden sisällölliset ja tyylimääritykselliset eroavaisuudet näiden omilla hakijoille tarkoitetuilla sivuilla. Muutoksissa täytyykin huomioida myös nämä, ettei yksittäisten järjestelmien ulkoasu muutu. Tällöin vältetään ei toivotulta lopputulokselta. Tämän vuoksi asetukset on hyvä viedä yksitellen tarkastaen jokaisen asiakkaan käytössä oleva järjestelmä, että muutokset sulautuvat näihin oikein.

Nykyinen LAURA™-palvelun tyylitiedosto on yli 3200 riviä pitkä ja tämän muuttaminen nykyisestä kokonaan Mobile First-lähtöiseksi voi olla kovin työläs prosessi. Mobile first sopii paremmin vasta aloitettavaan projektiin, jolloin aloitetaan työskentely puhtaalta pöydältä. Tässä tilanteessa, kun nykyinen olemassa oleva tyylitiedosto on jo hyvin pitkälle kehitetty tämä ei olekaan järkevää, vaan aloitamme responsiivisen suunnittelun kehittämällä nykyistä jo olemassa olevaa tyylitiedostoa.

Avoimien työpaikkojen listaus on toteutettu taulukkona, mutta muutoin taulukko mallista esitystapaa ei hakijasivuilla esiinny. Tämä muodostuu ongelmalliseksi pienellä selainikkunalla, jolloin sisällön luettavuus kärsii (Kuvio 8).

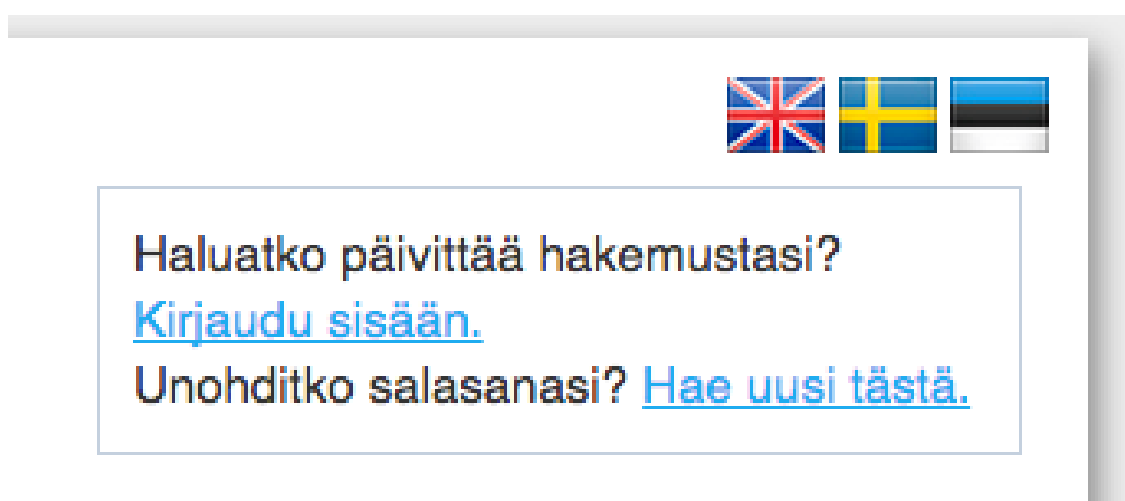
Avoimia työpaikkoja: 11

Tehtävän nimi	Hakuaika päättyy	Kaupunki	
Sijaisia lyhytaikaisiin sijaisuuksiin			Pääk

Kuvio 8: Nykyinen avoimien työpaikkojen listaus kavennetulla verkkoselaimella.

Nykyisin verkkosivuilla vieraileva ei näe avoimeen työpaikkaan liittyvää informaatiota taulukosta yhdellä silmäyksellä, vaan joutuu vierittämään tämän sisältöä sivuttaisessa suunnassa. Tämä tekee työpaikkojen luettavuudesta työläämpää. Taulukko voidaan purkaa mobiililaitteella työpaikkailmoitukset näyttämään ikään kuin ne olisivat omia elementtejä. Koska on hyvä suosia ensisijaisesti verkkosivun rakenteen muodostamiseen HTML- ja CSS-kieliä, jolloin tarkoituksena on toteuttaa taulukon purkaminen lähtökohtaisesti näillä kielillä. Kun toteutus on näiden osalta valmis ilman JavaScriptiä ja tämä näyttää jo näin hyvältä, voidaan ottaa JavaScript avuksi. Näin käyttö ei rajoitu myöskään niiltä, jotka eivät käytä JavaScriptiä.

Haasteeksi pienellä näytöllä muodostuu pienet painikkeet, joita on vaikea painaa sormenpäällä. Hakijasivulla on esimerkiksi pienet kielipainikkeet, joista voidaan vaihtaa sivujen kieltä toiselle kielelle. Näiden painaminen voi muodostua ongelmalliseksi älypuhelimella. Näiden kosketuspinta-alaa voisi suurentaa mobiililaitteella verkkosivuston käytettävyys huomioiden (Kuvio 9).



Kuvio 9: Nykyiset kielipainikkeet ja hakijan kirjautuminen.

Nykyisin hakijasivustoilla on käytössä ainoastaan kokomäärittämisistä pikseliarvoisia fontteja. Media queryn avulla voidaan käyttää aina erikokoisten laitteiden kohdalla erikokoisia fontteja, jolloin nämä voivat olla pikseleinä määritettyjä ja sivusto saataisiin käyttäytymään näillä responsiivisesti. Tämä olisi kuitenkin hyvin työlästä, koska tämä vaatisi aina laitekohtaisesti huomioida kaikki sivustoilla käytetyt fonttien kokojen määritykset. Kun käyttöön otetaan tämän tilalle em-yksiköt, tarvitsee määrittää ainoastaan pääfontin koko uudelleen. Pääfontin koko määrityksen sijaitessa body-tagissa, voidaan tämän arvoa muuttamalla vaikuttaa kaikkiin tämän mukaisesti skaalautuviin em-arvoihin määritettyihin fonttien kokoihin.

Fonttien kokojen yksiköistä vaihtoehtona responsiivisin olisi ottaa käyttöön vw-, vh-, vmin- ja vmax-yksiköt, jotka reagoivat heti selainikkunan koon muutoksiin. Kun viewport eli selainikkunan kokoon perustuvat fontit otettaisiin käyttöön, riittäisi lähtökohtaisesti sivustolla käytettyjen fonttien kokojen määritykset tehdä yhdestä paikasta. Kuitenkaan viewportin kokoon skaalautuvia fontteja ei vielä käytetä verkkosivuilla kovinkaan paljoa (Riethmuller 2016). Osittain tähän voi mahdollisesti vaikuttaa näiden vähäisempi tuki selaimilla, sillä Internet Explorer 8 ei tue tätä fonttityyppiä ollenkaan (Coyier 2012a). Vielä 2016 vuoden alussa Internet Explorerin käyttäjiä oli 1.18% muihin verkkoselaimiin verrattuna (Moss 2016). Tällä hetkellä kuitenkin maailmanlaajuisesti Can I Use (2017b) verkkosivuston mukaan käyttäjiä olisi kuitenkin enää vain 0.36 prosenttia. Toisaalta voidaan miettiä, että onko tarpeen enää tukea vanhoja selaimia, jotka muutenkin ovat jo käyttäjälle tietoturvan kannalta riskialttiina erilaisille verkkouhille (Kmetko 2016). Viewportiin skaalautuvat fontit voisikin olla mielenkiintoinen vaihtoehto ottaa käyttöön hakijasivuilla.

Uusista tekniikoista hilat eli grid layoutit vaikuttavat todella kätevilä. Hiloilla voisi sijoitella elementit asiakkaan halutessa asiakaskohtaisesti nopeasti käyttäen nimettyjä hiloja. Toinen hyöty olisi rekrytoijapuolella tämän käyttöliittymämainen ulkoasu, jolloin sisältö saataisiin tuntumaan enemmän järjestelmältä ja verkkosivumainen ulkoasu rikottua. Kuitenkin kyseessä on järjestelmä, jonka on toimittava kaikilla laitteilla luotettavasti myös ilman JavaScriptiä, jolloin tätä elementtien sijoittelutapaa ei voida ottaa vielä nykyisen tuen puitteissa käyttöön. Tulevaisuudessa tämä voisi kuitenkin olla toteutustapana vartenotettava.

CSS frameworkit tarjoavat helposti koko responsiivisen suunnittelun työkalut yhdessä paketissa. Hyötyinä tämän käyttöönottamisesta työnhakijapuolella olisi, että monet CSS frameworkit tarjoavat valmiita muun muassa pudostusvalikoita ja painikkeita, jotka ovat melko työläitä muokata aina asiakaskohtaisesti. Myös CSS frameworkin käyttöönottaminen voisi auttaa hallitsemaan responsiivisuutta, sillä tämä tarjoaa valmiin helpommin ymmärrettävän ruudukkorakenteen nykyiseen kellutukselliseen eli float tyyppiseen toteutukseen verrattuna. Kuitenkaan hakijasivu sisältää vain muutamia leveyssuunnassa

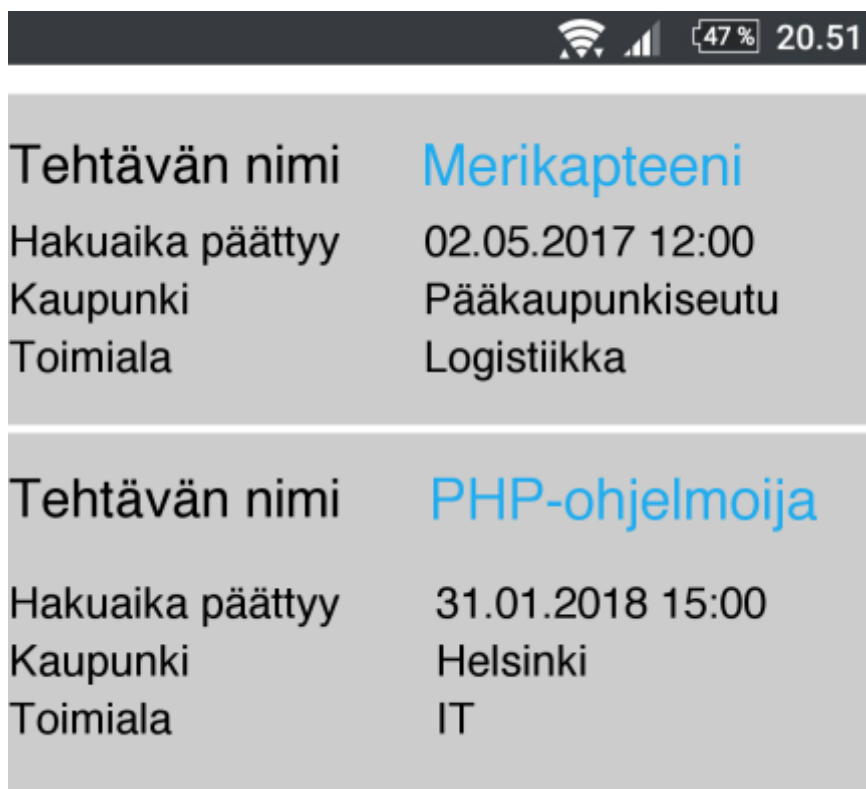
esitettäviä elementtejä, jolloin tämän tarjoama hyöty ei olisi kovinkaan suuri.

Työnantajapuolen järjestelmässä tästä voisi olla hyötyä, mikäli haluttaisiin kehittää tämän ulkoasua enemmän käyttöliittymä tyyppiseksi. CSS frameworkien tarjoamat grid systemit tarjoaisi tämän mahdollisuuden, kun ruudukko asetettaisiin käyttämään laitteen koko näytön leveydeltä tilan. Kuitenkin jo olemassa ja pitkään kehitetyn järjestelmän kanssa tämän käyttöönottoaminen tulee hyvin työlääksi, sillä nykyinen olemassa oleva rakenne täytyy silloin muuttaa toimimaan tämän pohjalta.

8 Responsiivisuuden implementoinnin suunnitelma LAURA™-palveluun

LAURA™-palvelun hakijasivuilla lähdetään kehittämään nykyisiä CSS-tyylimäärittelyjä hyödyntäen eri responsiivisen suunnittelun toteutustekniikoita, koska laitekohtaista mukautuvuutta on jo kehitetty aiemmin. Tärkeäksi hakijasivujen responsiivisuuden toteutuksessa muodostuu mediaehtojen eli media queryn käyttäminen. Näitä hyödyntämällä sivuston sisältö saadaan mukautumaan laitteen leveyden mukaan. Hakijasivuille lisätään puuttuva viewport-metatieto ja määritetään tämä käyttämään laitteen leveyttä, jonka lisääminen toimii lähtökohtana laitekohtaista mukautuvuutta sivustoille toteutettaessa. Fonttien joustavat kokomääritykset ovat myös tärkeä osa toteutettaessa responsiivisuutta LAURA™-palvelun hakijasivuille. Media queryn kanssa voidaan hyödyntää em- tai rem-yksiköitä pikseliarvojen tukena.

Avoimien työpaikkojen listauksen taulukkomallinen esitystapa voidaan purkaa mobiililaitteille, jotta informaatio tulee nopeasti ja helposti luettavasti esille. Kuviossa 10 on luonnos siitä, kuinka avoimien työpaikkojen lista voidaan esittää mobiililaitteella (Kuvio 10).



Tehtävän nimi	Merikapteeni
Hakuaika päättyy	02.05.2017 12:00
Kaupunki	Pääkaupunkiseutu
Toimiala	Logistiikka
Tehtävän nimi	PHP-ohjelmoija
Hakuaika päättyy	31.01.2018 15:00
Kaupunki	Helsinki
Toimiala	IT

Kuvio 10: Luonnos avoimien työpaikkojen listauksesta mobiililaitteella.

Tämän tapainen esitysmuoto selkeyttää luettavuutta pienikokoisilla laitteilla. Teksti on nopea lukea, koska nykyiseen taulukkomuotoiseen esitystapaan verrattuna tällä tavalla toteutettuna ei käyttäjän tarvitse suurennella näyttöä lähemmäksi. Käyttäjän ei myöskään ole tarpeen vierittää sivua, koska kaiken informaation voi lukea yhdellä silmäyksellä. Tämä tekee työilmoitusten selaamisesta helpompaa.

Nykyiset järjestelmän pienet painikkeet täytyy suurentaa, jotta nämä olisivat mobiililaitteilla helpommin käytettäviä. Käytössä olevat kieliä kuvaavat liput voidaan muuttaa tekstimuotoisiksi, jolloin näille saadaan helposti enemmän näkyvyyttä sekä enemmän kosketuspinta-alaa (Kuvio 11).



Kuvio 11: Hahmoitelma valikosta mobiililaitteella.

Painiketta voidaan painaa koko näytön leveydeltä. Kuitenkin jos asetukset ovat pysyvästi näkyvillä, voi tämä hidastaa sivun vieritystä alaspäin. Tämän korjaamiseksi, voidaan tieto piilottaa valikkoon. Valikkoa hahmottelevassa kuviossa 11 valikko on auki. Tämän saa auki painamalla valikkopainiketta.

9 Yhteenveto ja johtopäätökset

Opinnäytetyön tarkoitus oli selvittää keskeisimmät responsiivisen suunnittelun menetelmät. Opinnäytetyö on tutkimuksena kehittämistoiminnallinen, jonka myötä tämän opinnäytetyön tarkoitus on ensisijaisesti kehittää tutkimustuloksien pohjalta LAURA™-palvelun työnhakijoita palvelevien verkkosivustojen laitekohtaista mukautuvuutta responsiivista suunnittelua hyödyntäen. Tutkimustuloksien pohjalta onkin jo aloitettu responsiivisuuden kehittäminen LAURA™-palvelussa.

Nykypäivänä on jo miltein välttämätöntä käyttää verkkosivuilla responsiivista suunnittelua, mikäli haluaa tarjota palvelut myös ihmisten ollessa tien päällä. Erikokoisia verkkoa käyttäviä laitteita tulee kasvavissa määrin lisää, jonka myötä responsiivinen suunnittelu on tullut osaksi nykypäivää. Pienten, mukana kannettavien laitteiden käyttö on lisääntynyt ja kiireisen arjen keskellä esimerkiksi työpaikkoja voidaan selata julkisissa kulkuvälineissä tai vaikka kahvilassa. Puutteellisesti toimiva responsiivinen verkkosivu ei houkuttele käyttäjiä, vaan heikentää käyttömukavuutta ja siten vähentää kävijämääriä. On siis sanomattakin selvää, että responsiivisuus on tärkeä osa myös modernia rekrytointia, jossa otetaan huomioon nykyelämän realiteetit.

Responsiivisuuden toteutustapoja voisi sanoa olevan yhtä paljon, kuin web-kehittäjiäkin. On yrityksestä ja sen tarjoamista palveluista kiinni, millaiseen ratkaisuun päädytään ja mikä palvelee juuri kyseisiä tarpeita. Jokaiseen toteuttamistapaan liittyy toki omat vahvuutensa ja heikkoutensa, mikä tekee niiden vertailusta tärkeää. Verkkosivut voidaankin toteuttaa hyvin monella eri tavalla ja välttämättä ainoata ja oikeata tekniikkaa ei ole.

Responsiiviseen suunnitteluun käytettävät toteutuskeinot kehittyvät koko ajan. Tälläkin hetkellä on monia uusia tapoja sijoitella HTML-elementtejä, mutta nämä eivät ole vielä kuitenkaan kovinkaan vahvasti verkkoselaimilla tuettuina. Näillä pyritään ratkaisemaan monia nykyisiä haasteita sijoittelussa. Lisäksi kehityksen mukana tulee entistä pienempiä laitteita, jolloin sivustojen on kehityttävä vastaamaan myös näiden käyttäjien tarpeita.

Opinnäytetyön tutkimustulos vastaa tutkittua ilmiötä, jolloin tutkimus voidaan sanoa olevan validi. Koska tutkimus oli kehittämistoiminnallinen ja selvästi sijoittui tietyn ilmiön ympärille, saatiin tapaustutkimuksella tarpeellista tietoa. Tutkimus onnistui sijoittamaan tämän tiedon pohjilta saadut tutkimustulokset tutkittavaan ilmiöön. Tutkimukselliset tietolähteet perustuivat kirjallisiin ja sähköisiin tekstimateriaaleihin, jolloin sisällönanalyysillä saatiin validia tietoa tutkimusaineistoista.

Tutkimusmenetelmillä saatiin reliaabeliudeltaan luotettavaa tietoa, sillä sisällönanalyysiä hyödyntämällä tekstimuotoisista aineistoista saatiin analysoitua tutkitusta ilmiöstä

tutkimukseen soveltuvaa tietoa. Tapaustutkimuksen ja sisällönanalyysin avulla tutkimus onnistui kytkemään saadut tutkimustulokset ilmiön laajempaan kontekstiin.

Lähteet

Kirjalliset lähteet

Eskola, J. & Saarela-Kinnunen, M. 2010. Tapaus ja tutkimus = tapaustutkimus. Teoksessa Aalto, J. & Valli, R. (toim.) Ikkunoita tutkimusmetodeihin I: Metodien valinta ja aineistonkeruu: virikkeitä aloittelevalle tutkijalle. 3. Painos. Jyväskylä: PS-kustannus, 189-199.

Hampton-Smith, S. 2016. Pro CSS3 Layout Techniques. New-York : Springer Science+Business Media.

Heikkinen, H. L. T. 2010. NARRATIIVINEN TUTKIMUS - TODELLISUUS KERTOMUKSENA. Teoksessa Aalto, J. & Valli, R. (toim.) Ikkunoita tutkimusmetodeihin I: Metodien valinta ja aineistonkeruu: virikkeitä aloittelevalle tutkijalle. 3. Painos. Jyväskylä: PS-kustannus, 189-199.

Kadlec, T. 2013. IMPLEMENTING RESPONSIVE DESIGN: BUILDING SITES FOR AN ANYWHERE, EVERYWHERE WEB. Berkeley: New Riders.

Korpela, J. K. 2013. CSS : uudet mahdollisuudet. Jyväskylä: Docendo.

Kyrnin, J. 2015. Sams Teach Yourself Responsive Web Design in 24 Hours. Indianapolis : Pearson Education.

LaGrone, B. 2013. HTML5 and CSS3 Responsive Web Design Cookbook - Learn the secrets of developing responsive websites capable of interfacing with today's mobile internet devices. Birmingham: Packt Publishing.

Metsämuuronen J. 2005. Tutkimuksen tekemisen perusteet ihmistieteissä. 3 painos. Jyväskylä: Gummerus Kirjapaino.

Vilkka, H. 2005. Tutki ja kehitä. Helsinki: Kustannusosakeyhtiö Tammi 2005.

Sähköiset lähteet

Can I use. 2017a. CSS Grid Layout. Viitattu 4.4.2017. <http://caniuse.com/#search=grid>

Can I use. 2017b. Browser usage table. Viitattu 16.4.2017. <http://caniuse.com/usage-table>

Coyier, C. 2012a. Viewport Sized Typography. Viitattu 7.4.2017. <https://css-tricks.com/viewport-sized-typography/>

Coyier, C. 2012b. Dropdown Default Styling. Viitattu 15.4.2017. <https://css-tricks.com/dropdown-default-styling/>

Griffin, M. 2015. Readable Wearables. Viitattu 14.04.2017. <https://alistapart.com/column/readable-wearables>

Gube, J. 2013. 13 Best Responsive CSS Grid Systems for Your Web Designs. Viitattu 9.4.2017. <http://designinstruct.com/web-design/responsive-css-grid/>

Kharpal, A. Sony launches phone with world's first 4K HDR screen and superfast download speeds. Viitattu 6.4.2017. <http://www.cnbc.com/2017/02/27/sony-xperia-xz-premium-launch-smartphone-worlds-first-4k-hdr-screen-gigabit-download.html>

Kmetko, L. 2016. WHY YOU SHOULD STOP SUPPORTING IE10, IE9 AND IE8. Viitattu 16.4.2017. <https://www.xfive.co/blog/stop-supporting-ie10-ie9-ie8/>

- LAURA. Rekrytointijärjestelmä. Viitattu 19.3.2017 <http://laura.fi/rekrytointijarjestelma>
- Lennon, J. 2010. A guide to writing unobtrusive JavaScript and Ajax. Viitattu: 9.4.2017. <https://www.ibm.com/developerworks/library/wa-aj-unobtrusive/>
- Marcotte, E. 2010. Responsive web design. Viitattu 4.4.2017. <https://alistapart.com/article/responsive-web-design>
- Marcotte, E. 2011. Toffee-nosed. Viitattu 27.3.2017. <https://ethanmarcotte.com/wrote/toffee-nosed/>
- Moss, B. 2016. IE8 is back from the dead. Viitattu 16.4.2017 <https://www.webdesignerdepot.com/2016/01/ie8-is-back-from-the-dead/>
- Riethmuller, M. Truly Fluid Typography With vh And vw Units. Viitattu 16.4.2017. <https://www.smashingmagazine.com/2016/05/fluid-typography/>
- Rizzo, D. 2016. Accessible, Simple, Responsive Tables. Viitattu 11.4.2017. <https://css-tricks.com/accessible-simple-responsive-tables/>
- Saaranen-Kauppinen & Puusniekka. Reliabiliteetti. Viitattu 19.4.2017. http://www.fsd.uta.fi/menetelmaopetus/kvali/L3_3_2.html
- Smith, G. 2013. 20 Exceptional CSS Boilerplates and Frameworks. Viitattu 9.4.2017. <http://mashable.com/2013/04/26/css-boilerplates-frameworks/#zsyOMOd7Gqb>
- Snook, J. 2011. Font sizing with rem. Viitattu 16.4.2017. https://snook.ca/archives/html_and_css/font-size-with-rem
- Stephen, B. 2014. Pixel Practice in Web Design: The “Why?” Behind Virtual Pixels. Viitattu: 9.4.2017. <http://www.howdesign.com/featured/hardware-css-pixels-retina-display/>
- Tilastokeskus. 2015. Internetin käyttö mobiililaitteilla. Viitattu 17.3.2017. http://tilastokeskus.fi/til/sutivi/2015/sutivi_2015_2015-11-26_kat_002_fi.html
- Uranus Oy. 2017. Digitaalisen rekrytoinnin asiantuntija Uranus Oy uudistuu. Viitattu 14.4.2017. <http://rekrytointi.uranus.fi/uutishuone/uranus-oy-uudistuu/>
- Video4Change. The Basics on Video Resolution. Viitattu 14.4.2017. <http://www.v4c.org/en/content/basics-video-resolution>
- Wallace, W. 2015. Responsive Tables in Pure CSS. Viitattu 12.4.17. <https://techblog.livingsocial.com/blog/2015/04/06/responsive-tables-in-pure-css/>
- W3Schools, a. HTML Responsive Web Design. Viitattu 25.3.2017. https://www.w3schools.com/html/html_responsive.asp
- W3Schools, b. Responsive Web Design - Media Queries. Viitattu 25.3.2017. <https://www.w3.org/standards/webdesign/htmlcss>
- W3Schools, c. Responsive web design - grid view. Viitattu 2.4.2017. https://www.w3schools.com/css/css_rwd_grid.asp
- W3Schools, d. HTML <meta> name Attribute. Viitattu 12.4.2017. https://www.w3schools.com/tags/att_meta_name.asp
- W3. 2014. HTML & CSS. Viitattu 26.3.2017. <https://www.w3.org/standards/webdesign/htmlcss>

W3. 2017. CSS Grid Layout Module Level 1. Viitattu 2.4.2017. <https://www.w3.org/TR/css3-grid-layout/>

Ziemann, M. 2017. Työpaikkaa vaihtamassa? Töitä etsitään innokkaimmin maanantaisin ennen lounasta. Viitattu 12.4.2017. <http://yle.fi/uutiset/3-9466659>

Suulliset lähteet

Miettinen, P. 2017. Liiketoimintajohtajan haastattelu 12.4.2017. LAURA Rekrytointi Oy. Helsinki.

Kuviot

Kuvio 1: LAURA™:n demoversion hakijasivut mobiililaitteelta katsottuna.	8
Kuvio 2: Responsiivinen verkkosivu kuvattuna erilaisilla laitteilla.	11
Kuvio 3: Esimerkki nimetyistä ruudukko riveistä, vasemmalla CSS grid layout määitykset ja oikealla lopputulos.	13
Kuvio 4: CSS-mediaehdot sisäisessä ja ulkoisessa muodossa.	14
Kuvio 5: Viewportin määitykset.	15
Kuvio 6: em-yksikkö käytettynä fonttien koko määityksissä.	17
Kuvio 7: Nykyinen hakemuslomake kavennetulla verkkoselaimen ikkunalla.	21
Kuvio 8: Nykyinen avoimien työpaikkojen listaus kavennetulla verkkoselaimella.	22
Kuvio 9: Nykyiset kielipainikkeet ja hakijan kirjautuminen.	22
Kuvio 10: Luonnos avoimien työpaikkojen listauksesta mobiililaitteella.	25
Kuvio 11: Hahmoitelma valikosta mobiililaitteella.	26