



Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Juha Selonen

Responsiivinen mobiilisivusto CSS-ohjelmistokehyksellä

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknikka

Insinöörityö

29.5.2020

Tekijä Otsikko	Juha Selonen Responsiivinen mobiilisivusto CSS-ohjelmistokehyksellä
Sivumäärä Aika	36 sivua + 1 liite 29.5.2020
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintäteknikka
Ammatillinen pääaine	Mediateknikka
Ohjaaja	Lehtori Ulla Sederlöf
<p>Insinööriyön tarkoituksena oli tehdä Bootstrap-CSS-ohjelmistokehyksen avulla responsiivinen internetsivusto, joka on ensisijaisesti tarkoitettu käytettäväksi älypuhelimien näytöltä. Insinööriyössä perehdyttiin responsiiviseen suunnitteluun ja teknologioihin, joilla CSS-ohjelmistokehys on toteutettu. Tutkittuja teknologioita olivat HTML5 ja CSS. CSS:n osalta selvitettiin Flexbox- ja CSS Grid -teknologioita. Työssä tutkittiin responsiivisen suunnittelun filosofiaa. Niin sanottu mobiili ensin -suunnittelu lähtee pienimmästä näyttökoosta ja skaalautuu myös isoille näytöille. Työssä selvitettiin, miten suosikkikuvat rakennetaan mobiilissa ympäristössä käytettäväksi. Insinööriyössä tehtiin Bootstrap-ohjelmistokehyksen avulla internetsivut.</p> <p>Työssä tutkittiin, mikä on käytetyin CSS-ohjelmistokehys internetsivujen tekemiseen ja käytetyin kehittäjien keskuudessa. Suosituin CSS-ohjelmistokehys on Bootstrap, jota käytetään eniten internetsivujen tekemiseen. Myös kehittäjien keskuudessa se on suosituin. Työssä tutkittiin myös Materialize CSS -ohjelmistokehystä sen vuoksi, että työssä ei haluttu verrata kahta suosituinta, vaan erilaisia CSS-ohjelmistokehymiä. Muiden CSS-ohjelmistokehysten osalta käytetyimmät olivat Bootstrap, Foundation, UIkit, Materialize, Semantic UI ja Bulma.</p> <p>Insinööriyön lopputuloksena syntyi responsiivinen mobiilisivusto. Sivuston aihe on tekijän oman osaamisen esittely potentiaalisille työnantajille. Sivusto onnistui hyvin ja toimii mobiilissa ympäristössä. Insinööriyöraportissa esitellään, miten sivusto on suunniteltu ja toteutettu. Työssä esitellään sivuston rakenne ja layout-kaaviot sekä se, mihin tarkoitukseen sivusto on tehty ja mikä on sen kohdeyleisö. Jatkokehityksen kannalta käytettävyyteen ja visuaalisuuteen tulee vielä paneutua.</p>	
Avainsanat	CSS-ohjelmistokehys, responsiivisuus, Bootstrap

Author Title	Juha Selonen Responsive mobile website with CSS framework
Number of Pages Date	36 pages + 1 appendix 29 May 2020
Degree	Bachelor of Engineering
Degree Programme	Information and Communications Technology
Professional Major	Media Technology
Instructor	Ulla Sederlöf, Senior Lecturer
<p>The purpose of this thesis was to make a responsive website using Bootstrap CSS framework, primarily intended for use on a smartphone screen. The study combines the theoretical part with the practical implementation. In the theoretical part the theory of responsive design and the technologies that are used in the latest CSS frameworks were studied. The technologies studied were HTML5 and CSS. For CSS, flexbox and CSS grid technologies were investigated. The thesis explored the philosophy of responsive design. This thesis describes the design method which starts from the smallest screen size and scales to the large screens; this is called "mobile first"-design. The project also studied how to build favorite icons to be used in a mobile environment. Webpages were made using Bootstrap.</p> <p>This thesis studied the most used CSS framework for website creation and the most widely ones used among developers. The most popular CSS framework is Bootstrap, this is the one which is most used for creating web pages. This is also the most popular framework among developers. Materialize CSS Framework is also used because I did not want to compare the two most popular ones but different CSS frameworks as well. For the other CSS frameworks, the most used ones were Bootstrap, Foundation, Ulkit, Materialize, Semantic UI and Bulma.</p> <p>As a result of this thesis was a responsive mobile site was created. The topic of the site is to present my skills to potential employers. The site was built successfully and works in a mobile environment. The work explored how the site was designed and executed. This study presents the site structure and layout diagrams and also explains the purpose of the site and its target audience. In terms of future development more time has to be used to improve usability and visual appearance.</p>	
Keywords	CSS framework, responsive, Bootstrap

Sisällys

Lyhenteet

1	Johdanto	1
2	Responsiivinen suunnittelu	2
3	Responsiivisen sivuston teknologia	6
3.1	HTML5-kuvauskieli	6
3.2	CSS3-tyylimäärittely	9
3.3	CSS-ohjelmistokehys	14
3.4	Bootstrap-ohjelmistokehys	14
3.5	Materialize-ohjelmistokehys	16
3.6	Mobiili ensin -suunnitteluperiaate	17
4	Responsiivinen portfoliosivusto	19
4.1	Synopsis	19
4.2	Rakennekaavio ja layout	19
4.3	Favicons mobiilille	22
4.4	Resurssit	23
5	CSS-ohjelmistokehysten käyttöaste ja suosio	24
5.1	BuiltWith-työkalu	24
5.2	GitHub Stargazers -tähdet	25
5.3	Stackoverflow tags -arviot	26
5.4	Portfoliosivuston tulos	26
6	Yhteenveto	32
	Lähteet	33
	Liitteet	
	Liite 1. Portfoliosivu älypuhelimien näytöllä	

Lyhenteet

HTML	Hypertext Markup Language on verkkosivujen tekemiseen käytetty merkin- täkieli.
CSS	Cascading Style Sheets on verkkosivuja varten kehitetty tyyliohjeiden laji.
WHATWG	Web Hypertext Application Technology Working Group. HTML:n ja siihen liittyvien teknologioiden kehityksestä kiinnostuneiden henkilöiden yhteen- liittymä.
W3C	World Wide Web Consortium. Kansainvälinen yritysten ja yhteisöjen yh- teenliittymä, joka ylläpitää ja kehittää internetin standardeja tai suosituksia.
SASS	Syntactically Awesome Style Sheets. CSS:n esiprosessointia.
HTTP	Hypertext Transfer Protocol on protokolla, jota selaimet ja palvelimet käyt- tävät tiedonsiirtoon.
RWD	Responsive Web Design. Responsiivinen eli mukautuva suunnittelu.

1 Johdanto

Insinööriyön tarkoituksena on tehdä responsiivinen mobiilisivusto. Responsiivisuus tarkoittaa sitä, että sivut skaalautuvat eli mukautuvat käytetyn laitteen näytön kokoon. Mobiilisivustona tehdään tekijän oman osaamiseen esittelyyn tarkoitettu sivusto JuhaWeb. Sivusto tehdään Metropolian opinnäytetyötä varten, mutta se toimii tämän jälkeen oman osaamisen esittelynä ja tarkoitus on ylläpitää sitä aktiivisesti. Työn tarkoituksena on myös vertailla eri CSS-ohjelmistokehyksiä, niiden suosiota kehittäjien keskuudessa ja sitä, mitä niistä käytetään eniten eri sivustoilla. Teen tämän perusteella johtopäätöksen, mikä ohjelmistokehys soveltuu parhaiten responsiivisen mobiilisivuston toteuttamiseen. Valitsin lopuksi yhden CSS-ohjelmistokehyksen ja toteutin sivuston tällä ohjelmistokehyksellä.

Aluksi insinööriyössä tutkittiin, mitä CSS-ohjelmistokehyksiä käytetään vuonna 2020 (1; 2; 3; 4). Valitsin näistä kuusi eniten mainittua ja vertailin niiden suosiota. Suosiota mitattiin käyttäen Builtwith-sivuston (<https://pro.builtwith.com/newReport/technology>) teknologiaraporttityökalua. Vertailin ohjelmistokehysten suosiota kehittäjien parissa vertaamalla GitHub Stargasers- ja Stackoverflow-tageja.

Valitsin kaksi CSS-ohjelmistokehystä, joihin keskityn: Bootstrap ja Materialize. Valitsin Bootstrapin, jolla teen oman sivustoni. Valitsin Bootstrapin sen takia, että se on suosittu eri kehittäjien parissa ja sitä käytetään yleisesti eniten. Tämän vuoksi sillä on vakaa tulevaisuus ja sitä kehitetään aktiivisesti.

CSS-ohjelmistokehykset helpottavat sivuston kehitystä, koska niillä voidaan käyttää yhtenäistä CSS-rakennetta ja sivut näyttävät samoilta kaikilla eri selaimilla. Suosituin CSS-ohjelmistokehys on tällä hetkellä Bootstrap, ja sen uusin versio on 4.

2 Responsiivinen suunnittelu

Responsiivinen suunnittelu on kehitetty 2010-luvun alussa, kun mobiililaitteiden määrä kasvoi ja yhä useampi alkoi käyttää internetsivuja mobiililaitteilla (5). Vuonna 2016 mobiilikäyttäjien määrä ohitti pöytätietokoneiden käyttäjien määrän. Responsiivisen suunnittelun tavoitteena on, että internetsivut mukautuvat käytetyn laitteen ja näyttökoon mukaan. Yksi responsiivisen suunnittelun koulukunnista on mobiili ensin -ajattelu, jossa sivujen suunnittelu lähtee pienimmän näyttökoon lähtökohdista (37). Ennen sivustot tehtiin erikseen suurille näytöille ja mobiililaitteille, mutta tämä oli työlästä, kun samasta sivustosta piti tehdä monta eri versiota (5).

Ethan Marcotte käytti ensimmäisen kerran termiä responsive web design (RWD) arikkelissaan "A List Apart" vuonna 2010 (37). Responsiivinen suunnittelu koostuu kolmesta osasta: mukautuva suunnitelma (fluid layouts), mukautuvat kuvat ja media (flexible images) ja mediakyselyt (media queries). Responsiivinen suunnittelu ja mukautuva sisältö luo yhtenäisen standardin, jolla sivustot suunnitellaan. Responsiivinen suunnittelu on muutakin, kuin vain tekniset säännöt: se on joustava lähestymistapa siihen, miten suunnitella sivustot nyt ja tulevaisuudessa (5).

Mukautuva suunnitelma (fluid layouts) on suunnitelma, jossa sivuston sisältö mukautuu käytettävän laitteen ja näyttökoon mukaan ja sivulla olevat komponentit mukautuvat eli "kelluvat" sen mukaan. Mukautuvassa suunnitelmassa määritetään maksimi leveys (max-width) prosentteina. Prosenttimäärittely on tärkeä osa mukautuvaa suunnitelmaa (6).

Verkko (grid), johon mukautuvaa suunnitelmaa käytetään, koostuu riveistä ja sarakkeista. Yleensä sarakkeiden määrä on tasaluku, jotta sarakkeet ovat helpompi jakaa. Yleisimmät sarakkekoot ovat 12, 16 ja 24. Usein käytetyin on 12 saraketta (37).

Mukautuvassa suunnitelmassa määritetään maksimikoko suunnitelmalle. Tämä suunnitelma jaetaan tietynkokoisiin sarakkeisiin, jotta siitä saataisiin selkeä ja helposti hallittava. Kaikkien elementtien leveys ja korkeus määritetään suhteellisessa koossa. Tällä saavutetaan, se, että kun sivun koko muuttuu, kaikkien sivustolla olevien elementtien

leveys ja korkeus muuttuu suhteessa niille varattuun tilaan. Kaikissa CSS-ohjelmistokehyksissä on valmiina oma mukautuva suunnitelma (6).

Toinen responsiivisen suunnittelun osista on mukautuvat kuvat (flexible images). Internet sivujen teksti on oletuksena juoksevaa, ja se mukautuu näyttökoon muuttuessa. Kuvat eivät ole oletuksena responsiivisia, vaan ne näkyvät samankokoisena kaikilla eri näyttöko'oilla ja osa kuvasta pois leikkautuu, kun se ei enää mahdu sisällön sisään. Mobiiliselailulle tämä aiheuttaa omat ongelmansa, kun kuvat eivät skaalaudu oikein tekstiin nähden (7).

Yleensä CSS-ohjelmistokehyksillä on omat tapansa skaalata kuvat responsiivisiksi:

Yksi tapa tehdä kuvista responsiivisia on määrittää kuvien koko suhteellisilla mitoilla. Sen voi tehdä laittamalla kuvan maksimikooksi 100 % (7), kuten esimerkkikoodissa 1 on tehty.

```

```

Esimerkkikoodi 1. Responsiivisen kuvan määrittäminen.

Kuvat tällä tekniikalla skaalautuvat, kun niillä on tarpeeksi tilaa jäljellä siinä tilassa, missä ne ovat (7).

Tässä tekniikassa on muutama huono puoli: Isot kuvat, jotka ovat yli 420 pikseliä, vievät sitä enemmän tilaa sivulla, mitä pienemmäksi näyttökoko menee. Varsinkin mobiilissa kuvat näyttävät isommilta suhteessa tekstiin, koska natiivi koko on suurempi kuin leveys monissa älypuhelimissa. Kun sivua ladataan ensimmäistä kertaa, ei selain voi varata tilaa kuvalle, ja kun kuva ladataan, se näyttää ponnahtavan ruudulle. Jos kuva ei ole koko tilan kokoinen, huippuresoluutioiset ja retinakuvat näkyvät kaksi kertaa senlevyisinä, kuin ne ovat (7).

Toinen tapa tehdä kuvista responsiivisia on määrittää kuvan leveys prosentteina suhteessa sivun leveyteen. Esimerkiksi kun kuvan koko on 500 x 300 pikseliä ja dokumentin koko on 1200 pikseliä, kuvan tarvitsema tila on $(500 / 1200) \times 100 = 41,66 \%$. Tämä luku laitetaan kuvan leveydeksi (7), kuten esimerkkikoodissa 2 on tehty.


```

```

Esimerkkikoodi 2. Kuvan responsiivisuus.

Tällä tavoin toteutettuna kuva pysyy skaalassa testin kanssa. Tämä menetelmä toimii hyvin korkearesoluutioisten ja retinakuvien kanssa, mutta liian suurien tai pienien näyttökokojen kanssa kuva voi näyttää liian isolta pieneltä suhteessa tekstiin. Tämän takia kuvalle laitetaan maksimileveys (7).

Jos kuville määrittelee vain leveyden, saattaa tämä rasittaa selainten suorituskykyä, koska selaimen pitää päivittää kuvat aina, kun sivun kokoa muutetaan. Yleensä viiveet ovat millisekunnin luokkaa, mutta jos kuvia on useita ja monia muita skaalautuvia, ne kasaantuvat ja sivujen lataus hidastuu. Tämä voidaan ottaa huomioon määrittämällä korkeus samalla (7), kuten esimerkkikoodissa 3.

```

```

Esimerkkikoodi 3. Responsiivisen kuvan leveys ja korkeus määrittely.

Selaimet eivät ole hyvin optimoituja muuttamaan laadukkaiden kuvien kokoa. Etenkin bitmap-kuvat saattava kärsiä sumuisuudesta, kun niitä skaalataan pienemmiksi. CSS-asetus "webkit-optimize-contrast" voi auttaa (7) katso esimerkkikoodi 4.

```
img {
  image-rendering: -webkit-optimize-contrast;
}
```

Esimerkkikoodi 4. CSS webkit-optimize-contrast -asetus

Myös kuvien kokoa pitää muuttaa: vaikka kuvat esitetään skaalattuina, ne vievät silti aikaa ladata varsinkin mobiilikäytössä. Siksi on hyvä tehdä erikokoisia kuvia eri laitteille. Nämä koot ovat pieni 320 pikseliä leveä, keskikokoinen 480 pikseliä leveä ja suuri 768 pikseliä leveä. Kuvien muuttaminen kolmeen kokoon on kuitenkin melko työlästä varsinkin, jos kuvia on paljon. Sivuille pitää laittaa myös mediakysely jokaiselle kuvakoolle (37).

Mediakysely (media queries) on CSS-ominaisuus, joka on CSS3-versiossa. Mediakyselyn avulla pystytään määrittämään, milloin tiettyä CSS-määrittelyä käytetään. Mediakyselyiden avulla pystytään määrittämään uniikit CSS-määrittelyt, työpöytä-, tabletti- ja mobiilikoolle (8).

Mediakysely tehdään käyttämällä mediatyyppi ja määrittämällä sille ehtoja. Mediakysely on tosi, kun se mediatyyppi vastaa laitetta, millä sitä näytetään (37).

Mediatyypit, joita käytetään, ovat seuraavat:

- all: kaikille laitteille sopiva
- print: sopiva materiaaleille, jotka on sivutettu ja joita esikatsellaan ruudulta
- screen: käytetään kuvaruuduille
- speech: puhesyntetisaattorille.

Mediaominaisuudet ovat selaimen, laitteen tai ympäristön ominaisuuksia. Nämä ominaisuudet testaavat, onko ominaisuudella annettua arvoa, ja ovat vapaaehtoisia. Mediaominaisuudet laitetaan sulkumerkkien sisään. Mediakyselyssä voidaan käyttää loogisia operaatiota kuten NOT, ONLY ja AND (37).

Mediakyselyihin tarvitaan breakpoint, joka voi olla vaikka eri näyttökoko eri laitteella. Esimerkkikoodissa 5 on tehty mediakysely näyttökoolle, joka on 600 pikseliä.

```
@media only screen and (max-width: 600px) {  
  body {  
    background-color: blue;  
  }  
}
```

Esimerkkikoodi 5. Mediakysely.

Esimerkkikoodin 5 mediakyselyesimerkissä tehdään mediakysely näyttökoolle, joka on maksimissaan 600 pikseliä. Jos selaimen näyttökoko on 600 pikseliä tai pienempi, asetetaan taustaväriksi sininen (8).

3 Responsiivisen sivuston teknologia

3.1 HTML5-kuvauskieli

HTML 5 (Hyper Text Markup Language) on kuvauskieli, jota käytetään internetsivujen tekemiseen. HTML kuvaa dokumentin semanttisen rakenteen (9). Vuonna 2008 World Wide Web Consortium ja Web Hypertext Application Technology Working Group teki ensimmäisen HTML5-luonnoksen (10). Tämän hetken uusin versio on HTML 5.2 vuodelta 2017 (11).

Web-kehityksen ydin on HTML, se on koodi, josta jokainen internetsivu koostuu. Internetsivut eivät ole enää vain staattisia sivustoja, vaan niillä voidaan toteuttaa monimutkaisia sovelluksia, kuten Gmail. Tätä varten on kehitetty uusi standardi nimeltä HTML 5 (12).

Tim Berners-Lee kehitti ensimmäisen version HTML:stä vuonna 1989. Yhdistäminen ja yhteyden katkaiseminen lähteeseen olivat internetselaamisen suurimpia ongelmia. Berners-Lee kehitti kaksi teknologiaa, jotka korjasivat tämän: HTTP Hypertext Transfer Protocol ja HTML Hypertext Markup Language. HTTP on kuljetusprotokolla, jolla internet-palvelimet toimivat. HTML näyttää selaimessa sivuston käyttäjälle (12).

HTML käyttää tageja sivuston esittämiseen. Uusia tageja HTML:ssa ovat semanttiset tagit, jotka kuvaavat sivun rakennetta (12) ks. kuva 1.



Kuva 1. HTML5:n semanttinen rakenne (42).

HTML 5 sopii hyvin mobiilikäyttöön, ja tämä on hyvä, sillä yhä useampi käyttää älypuhelinia internetin selailuun. Älypuhelimessa on monia sisäänrakennettuja ominaisuuksia, joita HTML 5 pystyy hyödyntämään:

- GPS (Global positioning satellite)
- korkean resolution kamera
- videoiden nauhoitus
- kiihtyvyyssanturi
- gyroskooppi
- internetyhteys.

Mobiiliselainversiot käyttävät avoimen lähdekoodin järjestelmiä, joita kutsutaan nimellä WebKit. Googlessa on oma Webkit, jota se käyttää Chrome-selaimessa ja Android -laitteissa. Applella on oma Webkit, jota se käyttää Safari-selaimessaan ja iOS-laitteissaan. WebKitillä on suora tuki HTML5:lle (12).

Kaikilla mobiilivalmistajilla (Apple, Google, Microsoft, RIM ja HP) on nykyisin HTML5:tä tukeva selain. Applen ja Googlen Webkitien välillä ei ole suuria eroja: internetsivut toimivat molemmilla järjestelmillä (12).

Se, miten verkkosivujen suunnittelu mobiilikäyttöön eroaa tavallisista työpöydälle suunnatusta sivuista, on se, että näyttökoko täytyy ottaa huomioon. Älypuhelisten näyttökoko on rajoitettu: puhelimesta voi selata sivuja vaaka- ja pystysuunnassa, niissä on korkean resoluution näyttö, tekstinsyöttö tapahtuu kosketusnäytöltä (12).

Yleisimmät kuvakoot puhelimen näytölle vaihtelevat välillä 320 x 480 ja 960 x 640. Älypuhelisten näytön koolle on rajoitus: ihmisen käteen mahtuu vain tietyn kokoinen älylaite, ja jos siitä tulee liian iso, on sitä vaikea käyttää (12).

Älypuhelimilla sivuja voi selata sekä pysty- että vaakasuunnassa riippuen siitä, miten puhelinta pitää kädessään. Tämä asettaa omat haasteensa sivujen suunnitteluun: sivujen pitää olla joustavia ja mukautua kuvaruudun muutoksiin. Tässä eri CSS-ohjelmistokehykset ovat suureksi avuksi, sillä käsin tällaisen sivuston suunnittu voi olla työlästä (12).

Älypuhelisten resoluutio on korkealaatuinen Apple ja Google ovat omissa puhelimissaan ottaneet käyttöön korkearesoluutioiset kuvat. Tämä sen takia, että yleensä puhelinta pidetään 20 – 30 cm:n päässä kasvoista. Silmä näkee yksityiskohdat, jotka eivät välttämättä näy normaaliresoluutioisella näytöllä. Älypuhelisten huippunäyttöjen tarkkuus voi ylittää 240 DPI eli painolaatuista kuvaa älypuhelisten näytöltä. Nämä korkean resoluution kuvat ovat kuitenkin kooltaan suuria, mikä asettaa sivujen suunnittelulle omat haasteensa (12).

Älypuhelisten yleisin syöttölaite on käyttäjän omat sormet eli kosketusnäyttö. Sormen painallus ei ole tarkka, siksi esimerkiksi nappuloiden ja muiden kosketeltavien elementtien pitää olla tarpeeksi isoja. Applen käyttöliittymäsuunnitteluoppaan mukaan painikkeiden pitää olla vähintään 44 x 44 pikselin kokoisia. Pienimmällä 320 x 480 resoluution älypuhelisten näytöllä tämä pitää ottaa huomioon, kun suunnittelee internetsivuja (12).

Älypuhelisten valmistajat tukevat HTML5-standardia. HTML5:ssä on monia uusia ominaisuuksia, kuten video, audio ja canvas, joilla voidaan tehdä monipuolista multimediaa (12).

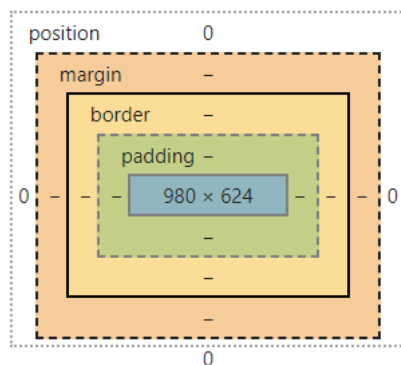
3.2 CSS3-tyylimäärittely

Cascading Style Sheets 3 on internetsivuilla käytettävä dokumentin tyylimäärittely. CSS3 perustuu box model-ideaan (kuvat 2 ja 3), joka koostuu seuraavista elementeistä:

- korkeus
- leveys
- marginaalit
- täyttö
- ääriiviivat (13).



Kuva 2. Box Model (43).



Kuva 3. Chrome inspector box model.

Toinen CSS:n tärkeä ominaisuus ovat kelluvat elementit (floating elements). Kelluvia laatikoita käytetään etenkin CSS-ohjelmistokehyksissä. Elementit voivat kellua oikealle tai vasemmalle tai ne voi pysäyttää kellumasta ollenkaan. Kelluvia elementtejä käytetään eniten ruudukoissa (grids) (13).

CSS:ssä on muutama näyttötyyppi (display type), jotka määrittävät miten elementti käytetään sivulla. Yleisimmät ovat seuraavat:

Block-elementti aloittaa aina uuden rivin. Tärkein ominaisuus on korkeus ja leveys. Yleisimpiä block-elementtejä:

- article
- aside
- div
- footer
- form
- h1, h2, h3, h4, h5, h6
- header
- main
- nav
- ol
- p
- table
- ul (13).

Inline-elementit vievät niin paljon tilaa, kuin ne tarvitsevat. Yleisimpiä inline-elementtejä:

- cite
- code
- dfn
- strong
- samp
- var
- a
- bdo
- br
- img
- map
- object
- script

- span
- sub
- sup
- button
- input
- label
- select
- textarea (13).

Inline-block-elementit ottavat ominaisuuksia inline- ja block-elementeistä. Inline-block voi määrittää korkeuden, leveyden ja marginaalin elementille. Inline-block ei aloita uutta rivinvaihtoa (13).

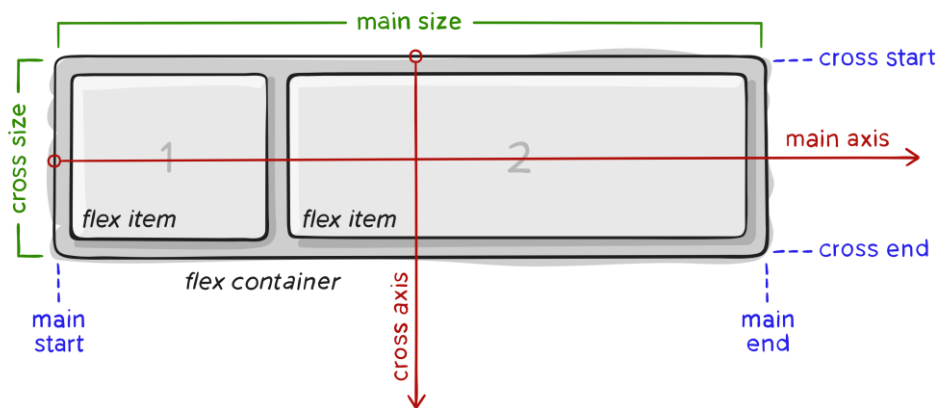
Kaksi perinteistä tapaa toteuttaa moderni responsiivinen sivusto ovat flexbox ja CSS grid.

Flexbox

Flexbox on tapa, jolla internetsivut määrittävät, miten niiden tila jaetaan, sisältö sijoitetaan ja miten visuaaliset elementit sijoittuvat toisiinsa nähden sivuilla. Flexbox-elementit järjestyvät pääakselin mukaan. Tämä akseli voi olla horisontaalinen tai vertikaalinen, eli elementit voidaan jakaa riveihin ja sarakkeisiin. Sisältö voi venyä, pienentyä, kääntyä toisinpäin ja uudelleen järjestyä sivuilla. Sitä voidaan myös määrittää pysty- ja vaakasuunnassa, sitä voidaan rivittää yhden tai useamman rivin mukaan tai sisältö voidaan mukauttaa käytettävän tilan mukaan. Flexboxin avulla sivuilla olevat elementit voidaan mukauttaa käyttäytymään eri lailla riippuen näyttökoosta, minkä ansiosta se sopii hyvin responsiivisten sivujen toteuttamiseen (38).

Flexbox käyttää vanhempi ja lapsi -suhdetta. Flexbox otetaan käyttöön määrittämällä "display: flex" elementille. Tästä elementistä tulee flex-säiliö, ja se järjestää käytettävissä olevan tilan kaikille sen lapsielementeille (38).

Flexbox toimii akseliverkkojärjestelmällä. Flexboxin avulla voi lisätä CSS-ominaisuuksien arvot elementtiin, joka määrittelee, miten lapsielementit tulisi asetella. Näitä elementtejä voi asetella vasemmalta oikealle, oikealta vasemmalle, ylhäältä alas ja alhaalta ylös. Elementtejä pystytään myös rivittämään vain yhdelle riville tai varaamaan tilaa sen verran, kuin elementille on käytettävissä kuten kuvassa 4 (38).

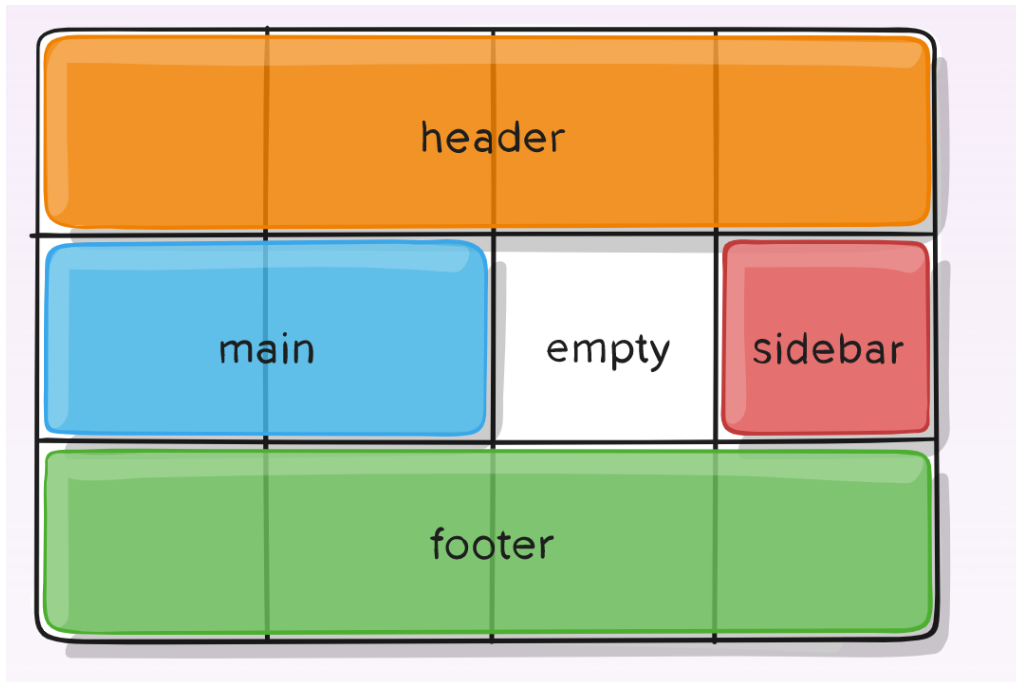


Kuva 4. Flexboxin toimintaperiaate (44).

Flexbox voi myös määrittää, miten tilaa, jota jää elementtien väliin, käytetään. Tilaa voi tasata vasemmalle, oikealle tai keskittää. Elementille voi myös määrittää, kuinka paljon tilaa eri lapsielementtien väliin jää ja miten ne ryhmitetään. Lapsielementeille voi määrittää, kuinka paljon tilaa ne vievät, tilaa voi varata yhdelle, useammalle tai kaikille elementeille. Elementeille voi myös määrittää, kuinka paljon tilaa ne vievät. Kaikille yhtä paljon tai voi määrittää kullekin elementille tietyn määrän. Eri elementtien järjestystä suhteessa toisiin elementteihin voi määrittää alas, ylös, keskelle tai venyttää viemään tilaa niin paljon, kuin elementillä on tilaa (38).

CSS Grid

CSS Grid -järjestelmän idea on jakaa internetsivu riveihin ja sarakkeisiin ks. kuva 5, joiden kokoa ja järjestystä pystytään määrittämään. Näitä ominaisuuksia pystytään määrittämään CSS:n avulla ilman, että HTML-koodiin täytyy tehdä mitään muutoksia. Media-kyselyitä voidaan hyödyntää määrittämällä pysäytyspisteitä eri näyttöko'oilte (39).



Kuva 5. CSS Grid -ruudukossa on rivejä ja sarakkeita (45).

CSS Grid on kaksisuuntainen järjestelmä, jossa tila jaetaan korkeuden ja leveyden mukaan. Flexbox on yksisuuntainen. CSS Grid lähtee suunnittelun näkökulmasta, kun taas Flexbox lähtee sisällön näkökulmasta. CSS Grid sopii hyvin suuriin ja monimutkaisiin sivustoihin (40).

3.3 CSS-ohjelmistokehys

CSS-ohjelmistokehykset perustuvat grid-systeemiin. Web-kehityksen yleinen ongelma oli CSS:n yhteensopivuus eri selainten kanssa. Yhteensopivuutta testataan ACID-testin avulla (14). ACID-testi mittaa, miten hyvin sivut noudattavat web-standardeja asteikolla 0 — 100. Tätä testiä ei enää pidetä täysin web-standardien mukaisena, ja sen ylläpito on loppunut vuonna 2013 (41). Sivut näyttivät erilaisilta eri selaimilla. CSS-ohjelmistokehys luotiin korjaamaan tämä tilanne luomalla yhtenäinen kokonaisuus eli ohjelmistokehys, joka toimii kaikilla eri selaimilla (14).

2000-luvun lopulla älypuhelimet yleistyivät ja internetsivuja alettiin katsoa enemmän pienemmiltä ruuduilta. Aiemmin mobiilia varten tehtiin omat sivustot, mutta CSS-ohjelmistokehysten ansiosta ei tarvitse tehdä erillisiä sivuja mobiilille (14).

Twitter Bootstrap julkaistiin vuonna 2011, ja se käytti kahdentoista sarakkeen responsiivista grid-asettelua. Samalla esiteltiin mobiili ensin -suunnittelun idea. Siinä oli tuki myös SASS:lle ja Flexboxille. SASS (Syntactically Awesome Style Sheets) on CSS:n esiprosessointia, jolla pyritään vähentämään toistoa tekemällä muuttujia ja lausekkeita (14).

3.4 Bootstrap-ohjelmistokehys

Bootstrap-ohjelmistokehityksen virallinen nimi on Twitter Bootstrap, ja se on alun perin kehitetty vuonna 2010 kahden Twitter-kehittäjän tekemänä. Ennen kuin siitä tuli avoimen lähdekoodin projekti, Bootstrap tunnettiin nimellä Twitter Blueprint. Kun siitä oli kehitetty muutama kuukausi, Twitter piti ensimmäisen hakkeriviikon, jolloin kaikenlaiset kehittäjät pääsivät kokeilumaan ja antamaan palautetta. Tätä palautetta käytettiin hyödyksi tyylilooppaisiin ja sisäisiin kehitystyökaluihin. Tätä samaa periaatetta käytetään vielä nykyäänkin Bootstrapin kehittämiseen. Bootstrapia kehittää pieni kehittäjätiimi GitHubissa, ja siitä on tullut yksi suosituimmista CSS-ohjelmistokehyksistä maailmassa (15).

Ensimmäinen versio julkaistiin vuonna 2011, ja sen jälkeen on julkaistu yli kaksikymmentä versiota. Suurimmat pääjulkaisut olivat versiot 2 ja 3 ja 4. Bootstrap 2 lisäsi responsiivisuuden koko ohjelmistokehykseen optionaalisena tyylitiedostona. Bootstrapin

versiossa 3 tätä responsiivisuutta kehitettiin edelleen ja otettiin käyttöön mobiili ensin-ajattelu (15).

Bootstrap 4:ssä oli kaksi suurta uudistusta: SASS otettiin käyttöön ja alettiin käyttää Flexboxia. Tämä helpotti uusimpien CSS-ominaisuuksien käyttöönottoa ilman ylimääräisiä riippuvuuksia. Bootstrap 4 tukee myös nykyisiä moderneja selaimia. V4.4.1 on tällä hetkellä uusin versio(15).

Bootstrap käyttää mukautuvaa suunnitelmaa, joka on jaettu 12 sarakkeeseen. Kaikki Bootstrapin sisällöt (container) kuva 6 ovat responsiivisia, ja Bootstrap käyttää mediakyselyitä ja responsiivisia breakpointeja (35).

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
<code>.container</code>	100%	540px	720px	960px	1140px
<code>.container-sm</code>	100%	540px	720px	960px	1140px
<code>.container-md</code>	100%	100%	720px	960px	1140px
<code>.container-lg</code>	100%	100%	100%	960px	1140px
<code>.container-xl</code>	100%	100%	100%	100%	1140px
<code>.container-fluid</code>	100%	100%	100%	100%	100%

Kuva 6. Bootstrap mukautuva suunnitelma (46).

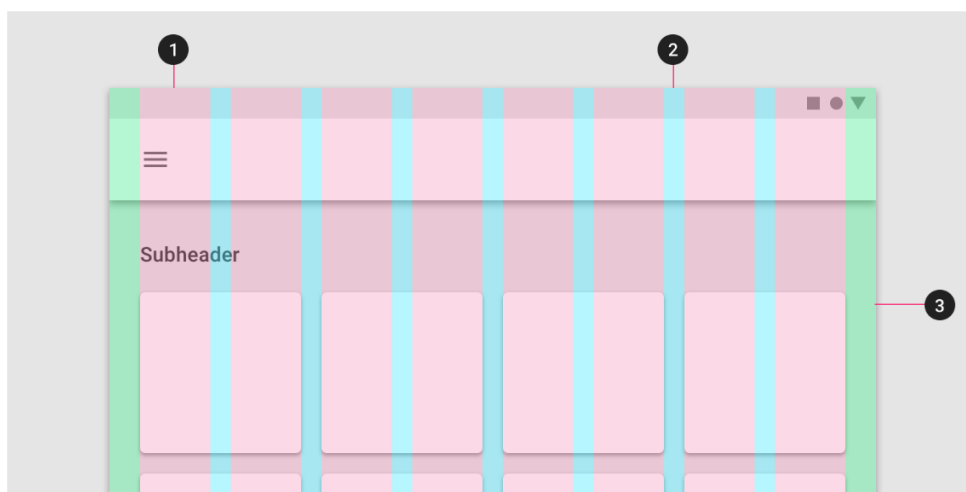
Bootstrapin ruudukko (grid) koostuu sisällöistä (container), riveistä (row) ja sarakkeista (columns), jotka pohjautuvat Flexbox-teknologiaan (36). Ruudukkoita on viiden kokoisia erittäin pienestä erittäin suureen.

3.5 Materialize-ohjelmistokehys

Materializen on suunnitellut ja luonut Google. Material Design on ohjelmistokehys, joka yhdistää klassiset suunnitteluperiaatteet innovaation ja teknologian kanssa. Materializen päämäärä on suunnittelualusta, jonka avulla saavutetaan yhtenäinen käyttäjäkokemus kaikilla tuotteilla ja alustoilla (16). The Material Design grid koostuu kolmesta elementistä kuva 7: riveistä (columns), välistä (gutter) ja reunasta (margin) (17).

Columns, gutters, and margins

The Material Design layout grid is made up of three elements: columns, gutters, and margins.



1. Columns
2. Gutters
3. Margins

Kuva 7. Material Design -asettelu. (17).

Sivujen sisältö laitetaan rivien sisään. Rivi määritetään prosentteina, jotta sisältö mahtuu mihin tahansa kuvakokoon (17).

3.6 Mobiili ensin -suunnitteluperiaate

Kumpi on parempi responsiivinen vai mobiili ensin -periaate? Tämä riippuu paljon kohdeyleisöstä, käyttääkö se sivuja mobiililaitteella vai pöytätietokoneella ja millainen on yhteysnopeus (18).

Responsiivinen suunnittelu lähtee desktopista: työpöytä on suurin koko ja sivut skaalautuvat siitä aina pienempään. Vaikka nämä sivut toimivat hyvin, mobiilinnavigaatio, sisältö ja latausnopeus on suunniteltu perinteisille internetsivuille (18).

Mobiili ensin lähtee siitä, että sivut on ensisijaisesti tarkoitettu mobiilille mutta toimivat myös tableteilla ja pöytätietokoneilla hyvin ilman suurempia muokkauksia. Sivuston asetelu ja suunnittelu pohjautuvat siihen, että se tarjoaa erinomaisen mobiilikäyttökokemuksen, sivut latautuvat nopeasti ja sisältö on monipuolinen ja kiinnostava, mikä herättää kohdeyleisön mielenkiinnon ja sivustoa on helppo käyttää kosketusnäytöltä (18).

Responsiivinen suunnittelu on yleisempää yrityksiltä yrityksille (B2B) -tapauksissa, kun, missä sisällön pitää olla informatiivista ja virallista. Näiden yritysten sivulla on HTML sisältöä, joka on jäsenelty hakukoneoptimointia varten. Suurin osa niiden asiakkaista käyttää sivuja kannettavalla tietokoneella tai työpöytäkoneella toimisto aikaan. Tämän suunnittelun mukainen rakenne ja käyttöliittymä toimii myös mobiilissa (18).

Responsiivisen suunnittelun hyviä puolia:

- sopii sivuille, joilla on paljon tietoa
- helpompi suurille lomakkeille ja monimutkaisille toiminnoille
- kustannustehokas suunnittelu ja ylläpito
- sopii hyvin hakukoneoptimointiin.

Responsiivisen suunnittelun huonoja puolia:

- mobiili käyttökokemus ei ole 100-prosenttisen optimoitu.

Yhä suurempi osa ihmisistä käyttää mobiililaitteita internetselailuun. Mobiili ensin -suunnittelu lähtee alusta alkaen mobiilista käyttökokemuksesta: mobiili natiivisovellusten kaltainen käyttöliittymä, vähemmän tekstiä, suurempi fonttikoko, nopea latausnopeus, ääni ja video, yhdelle sivulle vain yksi toiminto kerrallaan. Mobiili ensin -suunnittelussa voidaan käyttää myös puhelimen sisäänrakennettuja ominaisuuksia, kuten kameraa, tuntoaistiin perustuvaa palautetta ja ääni tunnistusta, ja niillä voidaan saavuttaa uniikki käyttökokemus (18).

Mobiili ensin -suunnittelun hyviä puolia:

- parempi käyttökokemus mobiililaitteilla
- suurin osa käyttää älypuhelinta internetselailuun
- voidaan käyttää puhelimen sisäänrakennettuja ominaisuuksia
- halvempaa kuin tehdä iOS-, Android- tai hybridisovellus.

Mobiili ensin -suunnittelun huonoja puolia:

- työpöytä käyttökokemus ei ole 100-prosenttisen optimoitu
- ei sovellu raskaalle sisällölle.

Jos halutaan valita joko mobiili ensin- tai responsiivinen suunnittelu, voidaan käyttää ns. 80–20-sääntöä. Jos 80 % kohdeyleisöstä käyttää suuria näyttöjä, käytetään responsiivista suunnittelua. Jos 80 % käyttää mobiililaitteita, käytetään mobiili ensin -suunnittelua. Mutta jos käyttö jakautuu tasan, niin analytiikkatyökaluilla, kuten Google Analytics, voi tehdä omia analyyseja siitä, mikä tapa sopii omalle sivustolle parhaiten (18).

4 Responsiivinen portfoliosivusto

4.1 Synopsis

Opinnäytetyön osana tehtiin responsiivinen mobiili portfoliosivusto, jolla esitellään tekijän omaa osaamista. Sivuston kohdeyleisö on yritys, joka tekee internetsivuja ja olisi valmis ottamaan vastavalmistuneen insinöörin töihin.

Sivuston laajuus käy ilmi kuvan 8 rakennekaaviosta: yksi pääsivu ja kolme alisivua. Sivusto sijaitsee domainhotelli.fi -palvelimella, ja sivustolla on oma rekisteröity domain ju-haweb.fi. Sivusto tehtiin Bootstrap -CSS-ohjelmistokehyksellä ja noudattaa HTML5-standardia. Sivusto suunniteltiin ensisijaisesti mobiiliympäristöön, mobiili ensin -suunnittelu -ajatuksella. Sivusto suojattiin SSL-tekniikalla ja sivustolla on Let's Encrypt -sertifikaatti.

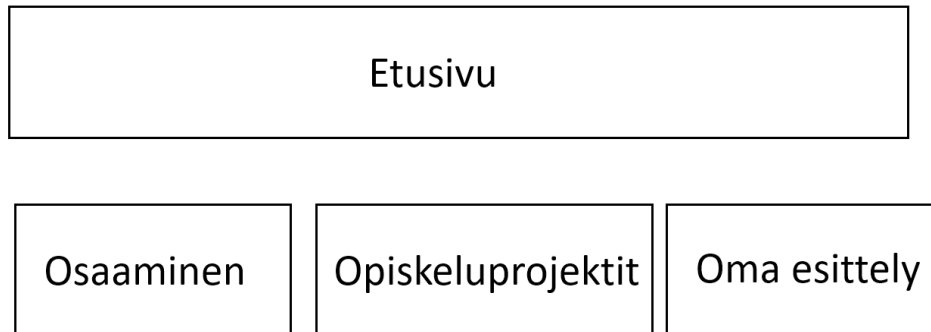
Käyttöliittymän suunnittelussa lähdettiin mobiili ensin -ajattelusta, jotta sivut sopisivat hyvin käytettäväksi mobiililaitteella mutta toimivat myös desktop-ympäristössä. Tavoitteena oli tehdä hyvät selkeät sivut.

Sivut päivitetään aina tarvittaessa, kun tulee uutta osaamista. Tietoturvapäivityksille ei ole tarvetta, koska, sivut eivät ole esim. WordPress-alustalla.

Työssä käytettiin omia materiaaleja. Tekijänoikeudet kaikista materiaalista on tarkistettu, että tekijänoikeudet ovat kunnossa.

4.2 Rakennekaavio ja layout

Portfoliosivuston rakenne on pääsivu, jolta pääsee kolmelle muulle sivulle (kuva 8). Etusivulla on lyhyt oma esittely ja linkit GitHub- ja LinkedIn-profiileihin.



Kuva 8. Portfoliosivuston rakennekaavio.

Etusivulta pääsee Osaaminen-sivulle, missä kerrotaan tekijän teknisestä osaamisesta ja on tietoa koulutuksesta ja työkokemuksesta. Opiskeluprojektit-sivuilla kerrotaan tärkeimmistä projekteista, jotka tekijä on tehnyt opintojensa aikana. Oma esittely -sivulla on vapaamuotoista kerrontaa tekijästä ja esimerkiksi harrastuksista.

Sivuston layout (kuva 9) on suunniteltu mobiili ensin -suunnitteluperiaatteen mukaan. Ylhäällä on sivuston header ja vasemmalla mobiilivalikko. Hero-osio on vain pääsivulla. Heron jälkeen tulee sivun sisältö. Viimeisenä on footer.



Kuva 9. Sivuston asettelun rautalankamalli.

Lähdin suunnittelemaan sivustoa ja rakennetta kuvan mukaisen asettelun rautalankamallin mukaisesti. Kuvassa on suunnitelma siitä, miltä sivut näyttävät, kun niitä katsotaan älypuhelimien näytöltä.

4.3 Favicons mobiilille

Sivustolle tehtiin mobiiliystävälliset faviconit, kuvakkeet, jotka näkyvät selaimen osoite-rivillä ja kun sivuston lisää kirjanmerkkeihin (kuva 10).



Kuva 10. Favicon selaimessa.

Suunnittelussa otettiin huomioon, että eri mobiiliympäristöissä pitää olla tietynlaiset kuvakkeet. Aluksi suunniteltiin selkeä J-kirjaimen kuva, jossa käytettiin samaa fonttia kuin itse sivustossa (kuva 11).



Brand Name

Icon Text

Icon Background

Icon Font Size

Font Family

Icon Font Color

Icon Background Color

Kuva 11. Favicon-asetukset.

Applen iOS-selaimelle laitettiin Apple Touch icon. Androidille luotiin oma kuvake ja sen kooksi asetettiin 192 x 192 pikseliä. Kuvakkeet tarkistettiin Favicon Checker (https://real-favicongenerator.net/favicon_checker) -työkalulla.

4.4 Resurssit

Aluksi mietittiin, mitä sisältöä sivuille voisi laittaa. Alkuperäisessä rakennekaaviossa oli monta ala sivua, mutta lopulta päädyttiin paljon yksinkertaisempaan ratkaisuun, jossa oli vain neljä elementtiä: etusivu, osaaminen, opiskeluprojektit ja oma esittely.

Layout käyttää Bootstrapin responsiivisuutta mobiili ensin -suunnitteluperiaatteella. Sivun yläosassa on header ja alaosassa footer.

Sisältö kirjoitettiin ensin Wordillä, jotta saatiin yleisimmät kirjoitusvirheet pois, ja sitten sisältö vietiin Brackets-editoriin ja muotoiltiin html-kielellä otsikoiksi ja kappaleiksi. Kaikki sivustolla käytetyt kuvat on itse otettu.

Sivut testattiin seuraavilla selaimilla:

- Chrome Versio 80.0.3987.122
- Firefox 73.0.1
- Microsoft Edge Versio 80.0.361.62.

Sivut testattiin myös Android-älypuhelimien Chrome 80 -selaimella.

Kaikki sivut validoitiin osoitteessa <https://validator.w3.org> W3C:n työkalulla. HTML5- ja CSS-validointi tarkistaa HTML:n ja CSS:n, että niissä ei ole syntaksivirheitä. Se tarkistaa, että tagit on suljettu oikein. Mitään virheitä ei löytynyt, vaan kakista tuli vastaus: "Document checking completed. No errors or warnings to show".

5 CSS-ohjelmistokehysten käyttöaste ja suosio

5.1 BuiltWith-työkalu

Selvitin BuiltWith-työkalun avulla, mitkä ovat tämän hetken käytetyimmät CSS-ohjelmistokehukset. BuildWithilla on tietojenanalysointijärjestelmä, joka on yli 11 vuoden ajan kerännyt lähes 100 % n kattavuudella aktiivisten verkkotunnusten ja aliverkkotunnusten tietoja. BuiltWith-työkalulla saadaan suuri määrä tietoa, joka on tilastollisesti merkittävä otos siitä, mitä teknologioita internetsivut käyttävät. Tämä työkalu käyttää uusimpia liikenteen tietolähteitä varmistaakseen, että se kattaa kaikki uusimmat sivustot, jotka ovat ajan tasalla 24 tunnin sisällä (19).

Jokainen internetsivu lähettää signaaleja siitä, millä teknologialla se on rakennettu. Järjestelmä voi seurata näitä signaaleja ja sen perusteella päätellä, mitä teknologiaa se käyttää. Järjestelmä pystyy seuraamaan kaikkia sivustoja, joilla on rekisteröity verkkotunnus (20).

Taulukkoon 1 on koottu yhteen, kuinka paljon eri CSS-ohjelmistokehystä käytetään internetsivuilla maailmanlaajuisesti.

Taulukko 1. CSS-ohjelmistokehystä käyttävien verkkosivustojen määrä maailmanlaajuisesti vuonna 2020 (47).

CSS-ohjelmistokehys	Kehystä käyttävien sivustojen määrä, kpl
Bootstrap	20 737 671
Foundation	508 999
UIkit	325 169
Materialize	119 533
Semantic UI	46 844
Bulma	32 079

Suosituin ohjelmistokehys on Bootstrap: sillä on tehty kaksikymmentämiljoonaa sivua. Muiden CSS-ohjelmistokehysten käyttö on paljon vähäisempää: toiseksi suosituimmalla

Foundation-kehyksellä on tehty puolimiljoonaa sivustoa. Bulma-kehyksellä on tehty vähiten sivuja maailmanlaajuisesti.

5.2 GitHub Stargazers -tähdet

GitHub on sivusto, jolla voi jakaa oman koodinsa muille käyttäjille ja työskennellä saman koodin kanssa monen eri kehittäjän kanssa. Koodit laitetaan repositorylle, joka on jokaisen projektin oma kansio, jossa on kaikki lähdekoodi ja ohjeet sekä tiedostojen historia (27).

GitHub Stargazer on ominaisuus, jonka avulla kehittäjät pystyvät antamaan repositorylle tähden, jos he ovat kiinnostuneet tästä teknologiasta. Tähden voi joko laittaa päälle tai pois. Muut käyttäjät voivat nähdä, mille teknologialle on annettu tähtiä ja onko heillä samanlaisia kiinnostusten kohteita (28).

Taulukossa 2 näkyy CSS-ohjelmistokehysten suosio kehittäjien keskuudessa.

Taulukko 2. GitHub-suosio: kuinka monta Stargazers-käyttäjää teknologialla on (21; 22; 23; 24; 25; 26).

CSS-ohjelmistokehys	Stargazers-käyttäjien määrä
Bootstrap	138 576
Materialize	47 311
Bulma	38 482
Semantic UI	37 102
Foundation	28 490
Ulkit	15 369

Eniten tähtiä oli annettu Bootstrapille. Vähiten suosituin oli Ulkit.

5.3 Stackoverflow tags -arviot

Stackoverflow on suosittu kehittäjien keskustelusivusto, jolla voi kysyä ja saada vastauksia koodausaiheisiin kysymyksiin. Tällä sivustolla on myös ominaisuutena tagit, joilla voi merkitä ja luokitella samankaltaisia kysymyksiä. Stackoverflow -agit ovat mittari, kuinka suosittu jokin teknologia on kehittäjien keskuudessa (29).

Taulukosta 3 selviää, että suosituin CSS-ohjelmistokehys on Bootstrap. Vähiten suosittu on Ulkit.

Taulukko 3. Kuinka monta Stackoverflow-tagia teknologialla on (20; 30; 31; 32 ;33; 34).

CSS-ohjelmistokehys	Stackoverflow tags -määrät
Bootstrap	97 139
Foundation	51 12
Materialize	3093
Semantic UI	2420
Bulma	518
Ulkit	187

5.4 Portfoliosivuston tulos

Insinööriyössä valittiin teknologiaksi Bootstrap, koska tarkoituksena oli tehdä responsiivinen sivusto käytettäväksi mobiilille. Sivuston toteutus onnistui hyvin: sivusto toimii selkeästi ja skaalautuu ruutukoon mukaan. Sivulla on selkeä header ja footer -osio, jotka näkyvät mobiilinäkymässä hyvin.

Sivustolla on selkeä suosikkikuvake (favicon), joka näkyy Android -ja ios-ympäristössä niiden omien määrittysten mukaan. iOS Safari-järjestelmälle on Apple Touch -kuvake ja Android-laitteille määriteltiin 192 x 192 pikselin levyinen kuvake ja Web App -manifesti.

Web App -manifesti on tiedosto, joka sisältää tietoja sivustosta. Tässä tapauksessa siinä on tiedot, mistä suosikkikuvake löytyy. Esimerkkikoodissa 6 on määritelty sivun Web App -manifesti.

```
{
  "name": "",
  "short_name": "",
  "icons": [
    {
      "src": "/android-chrome-192x192.png",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "/android-chrome-256x256.png",
      "sizes": "256x256",
      "type": "image/png"
    }
  ],
  "theme_color": "#ffffff",
  "background_color": "#ffffff",
  "display": "standalone"
}
```

Esimerkkikoodi 6. Portfoliosivuston Web App -manifesti.

Sivuston valikko toteutettiin Bootstrap-hampurilaisvalikolla, joka sijaitsee vasemmalla. Valikko avaa listan sivusta. Valikko on selkeä ja helppo käyttää mobiililaitteilla.

Kuvat muunnettiin responsiivisiksi Bootstrap-CSS-ohjelmistokehyksen avulla, ja ne skaalautuvat mobiilinäkymässä hyvin tekstiin nähden. Teksti muotoiltiin Bootstrap-marginaaleilla ja välistyksellä helppolukuseksi mobiilinäytöltä.

Esimerkkikoodissa 7 on osa etusivun lähdekoodista.

```
<!-- Sivun sisältö -->
<div class="container py-2 sisalto">
  <div class="m-3 p-3">
    <h2>Junior Web Developer</h2>
    <p>Opiskelen Insinööriksi tieto- ja viestintätekniiikan tutkinto-ohjelmassa Metropoliaassa. Suuntautumisvaihtoehtoni on mediatekniiikka.</p>
    <p>Tein opintoihini kuuluvan työharjoittelun Agenda Helsinki nimisessä yrityksessä. Työharjoittelu kesti 5kk.</p>
    <p>Olen tällä hetkellä tekemässä opinnäytetyötäni.</p>
```



```

    <p>Minulla on puolen vuoden työkokemus WordPress kehityksestä, tein
    työharjoittelussa internet-sivuja WordPressillä. Olen suorittanut itsenäisen
    WordPress kurssin, jossa oli WP perusteita ja tehtiin oma teema. Olen ollut
    puolen vuoden työkokeilussa, ja tein nettisivuja Drupal sisällönhallintajär-
    jestelmällä.</p>
    <h2>Github</h2>
    <p><i class="fab fa-github"></i> Github profiili : <a
    href="https://github.com/juhaselo" tar-
    get="_blank">https://github.com/juhaselo</a></p>

    <h2>Linkedin</h2>
    <p><i class="fab fa-linkedin"></i> Linkedin profiili : <a
    href="https://www.linkedin.com/in/juhaselonen" tar-
    get="_blank">https://www.linkedin.com/in/juhaselonen</a></p>
    <p>
    Olen tehnyt koulussa erilaisia projekteja, josta tarkemmin <a
    href="koulu.html">Opiskeluprojektit</a> osiossa. Tässä lyhyt yhteenveto:</p>
    <ul>
    <li>Kuvapalvelu sovellus</li>
    <li>Delicious Ruoka resepti sovelluksen</li>
    <li>Allstream Video streamaus sivuston</li>
    <li>Älybussi sovellus</li>
    <li>Valhalla Peli</li>
    <li><a href="https://metropolia.juhaweb.fi/" target="_blank">Oma
    WordPress teema</a></li>
    <li><a href="https://juhaweb.fi/godo/" target="_blank">Godo net-
    tisivut</a></li>
    </ul>
    </div>
</div>
<!-- Footer -->

<div class="card">

    <div class="card-body">
    <blockquote class="blockquote mb-0">
    <footer class="footer text-center">
    <p>JuhaWeb 2020.</p>
    </footer>
    </blockquote>
    </div>
</div>

```

Esimerkkikoodi 7. Portfoliosivuston etusivu toteutettiin Bootstrap CSS-ohjelmistokehyksellä.

Sivusto toteutettiin kokonaan Bootstrapilla. Koodiesimerkissä 7 on toteutettu etusivu. Etusivun sisältö laitettiin yhteen Bootstrap-säiliön sisään, ja sille määriteltiin valikko, hero, sisältö ja footer. Kuvassa 12 on portfoliosivun näkymä mobiilikoossa.

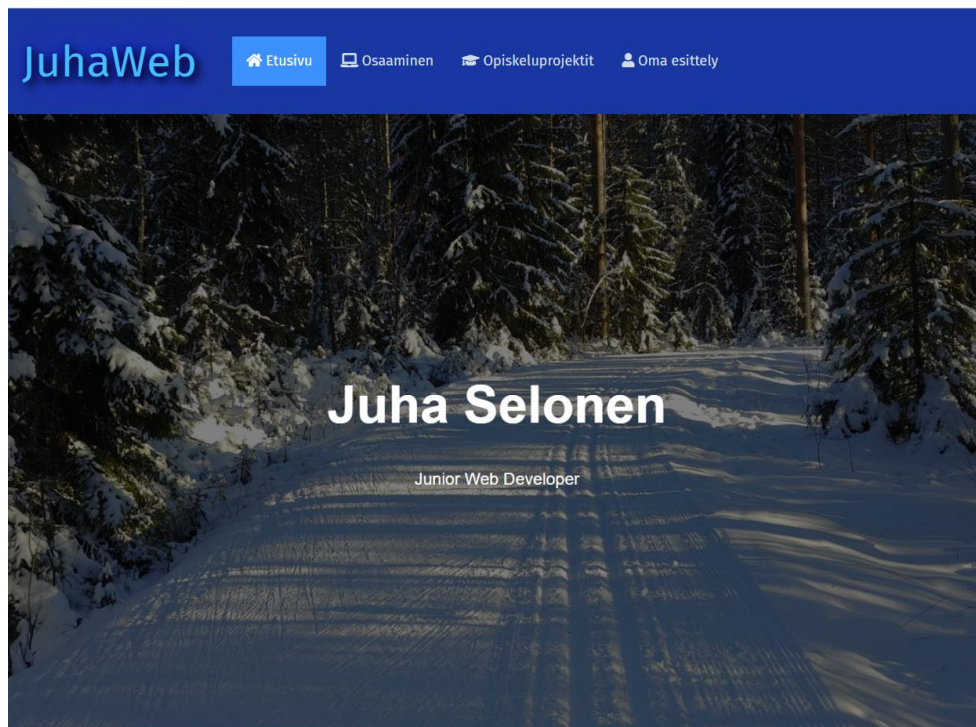


Kuva 12. Portfoliosivuston mobiilinäkymä.

Sivusto suunniteltiin normaalikoossa katsottuna suurimmillaan katsottavaksi 1920 x 1080 pikselin koossa. Käytävissä oli myös 4k-näyttö, ja sivusto testattiin 3840 x 2160 pikselin koossa. Desktop-näytöllä (kuva 13) sivuston logo on vasemmalla ja valikkorivi on tästä oikealla. Valikossa on animoitu efekti, kun hiirtä viedään yksittäisen valikon päälle ja pois. Sivustolla on selkeä suosikkikuvake, joka toimii kaikilla selaimilla. Kuvake on myös favicon.ico -muodossa.

Valikon jälkeen tulee hero-osio, jossa on taustakuva ja tekijän nimi ja tehtävänimike. Tämä osuus on vain etusivulla. Tämän jälkeen tulee varsinainen sivun sisältö. Viimeisenä elementtinä on footer.

Kuvat toteutettiin desktop-näkymässä responsiivisina Bootstrap-ohjelmistokehyksellä. Kuvat skaalautuvat näyttökoon mukaan.



Junior Web Developer

Opiskelen Insinööriksi tieto- ja viestintäteknikan tutkinto-ohjelmassa Metropoliasa. Suuntautumisvaihtoehdoni on mediatekniikka.

Tein opintoihini kuuluvan työharjoittelun Agenda Helsinki nimisessä yrityksessä. Työharjoittelu kesti 5kk.

Olen tällä hetkellä tekemässä opinnäytetyötäni.

Minulla on puolen vuoden työkokemus WordPress kehityksestä, tein työharjoittelussa internet-sivuja WordPressillä. Olen suorittanut itsenäisen WordPress kurssin, jossa oli WP perusteita ja tehtiin oma teema. Olen ollut puolen vuoden työkokeilussa, ja tein nettisivuja Drupal sisällönhallintajärjestelmällä.

Github

🔗 Github profiili : <https://github.com/juhaselo>

LinkedIn

🌐 LinkedIn profiili : <https://www.linkedin.com/in/juhaselonen>

Olen tehnyt koulussa erilaisia projekteja, josta tarkemmin Opiskeluprojektit osiossa. Tässä lyhyt yhteenveto:

- Kuvapalvelu sovellus
- Delicious Ruoka resepti sovelluksen
- Allstream Video streamaus sivuston
- Alybussi sovellus
- Valhalla Peli
- Oma WordPress teema
- Godo nettisivut

JuhaWeb 2020.

Kuva 13. Portfoliosivuston desktop-näkymä.

Jatkokehityksen kannalta aika, joka oli sivuston tekoon käytettävissä, riitti ainoastaan tekniseen suunnitteluun. Tulevaisuudessa jatkokehityksessä sivuston käytettävyyteen voisi tehdä parannuksia. Toinen jatkokehitysidea on visuaalisen ilmeen muokkaaminen: esimerkiksi värejä ja fontteja voisi miettiä.

6 Yhteenveto

Insinööriyössä tutkittiin, mikä on hyvä CSS-ohjelmistokehys responsiivisen verkkosivuston toteuttamiseen. Työssä tehtiin oman osaamisen esittelysivusto, joka toteutettiin Bootstrap-CSS-ohjelmistokehyksellä. Kun tutkittiin eri CSS-ohjelmistokehyksiä, suosituin kehittäjien ja käytön suhteen oli Bootstrap. Tämä ohjelmistokehys on yleisin internetsivuilla käytetty ohjelmistokehys BuildWithin mukaan. Myös kehittäjät suosivat sitä: suosio näkyy GitHub- ja StackOverflow-tilastoissa.

Tärkeimpiä teknologioita CSS-ohjelmistokehityksen taustalla ovat HTML5 ja CSS, jotka näyttävät kaikki visuaaliset elementit näytöllä. Bootstrap on tehty käyttäen Flexbox-tekniikkaa. Siinä sivu on jaettu yksisuuntaiseen tilaan ja sisältö näytetään sivulla Flexbox-määrittysten mukaan. Toinen responsiivisen suunnittelun teknologia on CSS Grid. Se on uudempi ja perustuu korkeus- ja pituussuunnassa toimivaan verkkoon.

Insinööriyön osana valmistui tekijän omaa osaamista esittelevä portfoliosivu, joka on responsiivinen ja toimii hyvin mobiiliympäristössä. Jatkokehityksen kannalta sivuston käytettävyyttä ja visuaalista ilmettä voisi vielä kehittää.

Lähteet

- 1 Top 6 popular CSS frameworks to use in 2020. 2020. Verkkoaineisto. Medium. <<https://medium.com/@liltuchi55/top-6-popular-css-frameworks-to-use-in-2020-d56fa7917d48>>. 4.1.2020. Luettu 12.2.2020.
- 2 Lazaris, Louis. 2019. 10 best CSS frameworks in 2020. Verkkoaineisto. <<https://www.creativebloq.com/features/best-css-frameworks>>. 10.9.2019. Luettu 12.2.2020.
- 3 Liu, Trista. 2019. 15 Best CSS Frameworks for Developers in 2020. Verkkoaineisto. <<https://www.mockplus.com/blog/post/css-framework>>. 2.12.2019. Luettu 12.2.2020.
- 4 Musienko, Yuri. 2019. 6 Incredible CSS Frameworks to Use in Web Development 2020 to Speed Up Your Website. Verkkoaineisto. <<https://merehead.com/blog/trends-css-frameworks-2020/>>. 27.12.2019. Luettu 12.2.2020.
- 5 Chilson, Joe. 2015. The evolution of responsive web design: a very brief history. Verkkoaineisto. <<https://www.1digitalagency.com/the-evolution-of-responsive-web-design-a-very-brief-history/>>. 26.3.2015. Luettu 16.3.2020.
- 6 Fluid Grid Layout Tutorial for Responsive Web Design. Verkkoaineisto. 1stWeb-Designer. <<https://1stwebdesigner.com/fluid-grid-layout/>>. 29.5.2016. Luettu 16.3.2020.
- 7 CSS Fluid Image Techniques for Responsive Site Design. Verkkoaineisto. The new code. <<http://thenewcode.com/586/CSS-Fluid-Image-Techniques-for-Responsive-Site-Design>>. Luettu 16.3.2020.
- 8 CSS Media Queries. Verkkoaineisto. CodeSpot. <<https://www.codespot.org/css-media-queries/>> Luettu 17.3.2020.
- 9 HTML Standart. Verkkoaineisto. Web Hypertext Application Technology Working Group. <<https://html.spec.whatwg.org/multipage/introduction.html>> Luettu 2.3.2020.
- 10 A brief history of HTML5. Verkkoaineisto. Html5 Marketplace. <<http://html5marketplace.com/briefhistoryHTML5.shtml>> Luettu 2.3.2020.
- 11 HTML 5.2. Verkkoaineisto. World Wide Web Consortium. <<https://www.w3.org/TR/html52>>. Luettu 2.3.2020.
- 12 Matthew, David. 2013. HTML5. O'Reilly Safari Online.

- 13 Sikora, Piotr. 2016. Professional CSS3. Packt Publishing.
- 14 Wanyoike, Michael. 2018. History of front-end frameworks. Verkkoaineisto. <<https://blog.logrocket.com/history-of-frontend-frameworks/>> 16.10.2018. Luettu 2.3.2020.
- 15 About Bootstrap. 2020. Verkkoaineisto. Bootstrap team. <<https://get-bootstrap.com/docs/4.4/about/overview/>> Verkkoaineisto. Luettu 2.3.2020.
- 16 About Materialize. Verkkoaineisto. Materialize. <<https://materializecss.com/about.html>> Luettu 2.3.2020.
- 17 Responsive layout grid. Verkkoaineisto. Google. <<https://material.io/design/layout/responsive-layout-grid.html#columns-gutters-margins>> Luettu 3.3.2020.
- 18 Choose your strategy: Mobile-First Web Design vs. Responsive Web Design. Verkkoaineisto. Darwin Digital. <<https://darwindigital.com/mobile-first-versus-responsive-web-design/>> Luettu 2.3.2020
- 19 Kramer, Ross. Data Coverage. Verkkoaineisto. <<https://builtwith.com/data-coverage/>>. 26.2.2020. Luettu 26.2.2020.
- 20 Frequency Asked Questions about BuiltWith. Verkkoaineisto. BuiltWith. <<https://builtwith.com/faq/>>. 26.2.2020. Luettu 26.2.2020.
- 21 Bootstrap Stargazers. Verkkoaineisto. GitHub. <<https://github.com/twbs/bootstrap/stargazers>>. 12.2.2020. Luettu 12.2.2020.
- 22 Materialize Stargazers. Verkkoaineisto. GitHub. <<https://github.com/dogfalo/materialize/stargazers>>. 12.2.2020. Luettu 12.2.2020.
- 23 Bulma Stargazers. Verkkoaineisto. GitHub. <<https://github.com/jgthms/bulma/stargazers>>. 12.2.2020. Luettu 12.2.2020.
- 24 Semantic Stargazers. Verkkoaineisto. GitHub. <<https://github.com/SemanticOrg/Semantic-UI/stargazers>>. 12.2.2020. Luettu 12.2.2020.
- 25 Foundation Stargazers. Verkkoaineisto. GitHub. <<https://github.com/foundation/foundation-sites/stargazers>>. 12.2.2020. Luettu 12.2.2020.
- 26 Uikit Stargazers. Verkkoaineisto. GitHub. <<https://github.com/uikit/uikit/stargazers>>. 12.2.2020. Luettu 12.2.2020.

- 27 Be Social. Verkkoaineisto. GitHub. <<https://guides.github.com/activities/socialize/>>. 17.4.2.2018. Luettu 26.2.2020.
- 28 Tags. Verkkoaineisto. GitHub. <<https://stackoverflow.com/tags>>. 12.2.2020. Luettu 12.2.2020.
- 29 Questions tagged [twitter-bootstrap]. Verkkoaineisto. Stackoverflow. <<https://stackoverflow.com/questions/tagged/twitter-bootstrap>>. 12.2.2020. Luettu 12.2.2020.
- 30 Questions tagged [zurb-foundation]. Verkkoaineisto. Stackoverflow. <<https://stackoverflow.com/questions/tagged/zurb-foundation>>. 12.2.2020. Luettu 12.2.2020.
- 31 Questions tagged [materialize]. Verkkoaineisto. Stackoverflow. <<https://stackoverflow.com/questions/tagged/materialize>>. 12.2.2020. Luettu 12.2.2020.
- 32 Questions tagged [semantic-ui]. Verkkoaineisto. Stackoverflow. <<https://stackoverflow.com/questions/tagged/semantic-ui>>. 12.2.2020. Luettu 12.2.2020.
- 33 Questions tagged [bulma]. Verkkoaineisto. Stackoverflow. <<https://stackoverflow.com/questions/tagged/bulma>>. 12.2.2020. Luettu 12.2.2020.
- 34 Questions tagged [getuikit]. Verkkoaineisto. Stackoverflow. <<https://stackoverflow.com/questions/tagged/getuikit>>. 12.2.2020. Luettu 12.2.2020.
- 35 Overview. Verkkoaineisto. Bootstrap team. <<https://getbootstrap.com/docs/4.4/layout/overview/>>. Luettu 18.3.2020
- 36 Grid system. Verkkoaineisto. Bootstrap team. <<https://getbootstrap.com/docs/4.4/layout/grid/>>. Luettu 18.3.2020.
- 37 Ward, Chris. 2017. Jump Start Responsive Web Design, 2nd Edition. SitePoint.
- 38 Weyl, Estelle. 2017. Flexbox in CSS. O'Reilly Media.
- 39 Kumar, Nitish & Adrian, Roworth & Ahmad Ajmi. 2017. Designing with CSS Grid Layout. SitePoint.
- 40 Gupta, Ayush. 2018. Beginner's Guide to CSS Grid and Flexbox. Verkkoaineisto. <<https://medium.com/youstart-labs/beginners-guide-to-choose-between-css-grid-and-flexbox-783005dd2412>>.15.6.2018. Luettu 24.3.2020.

- 41 Hickson, Ian. Acid Tests. Verkkoaineisto. <<https://www.acidtests.org/>>. Luettu 1.4.2020.
- 42 HTML Semantic Elements. Verkkoaineisto. W3schools. <https://www.w3schools.com/html/html5_semantic_elements.asp>. Luettu 15.4.2020.
- 43 CSS Box Model. Verkkoaineisto. W3schools. <https://www.w3schools.com/css/css_boxmodel.asp>. Luettu 15.4.2020.
- 44 A Complete Guide to Flexbox. Verkkoaineisto. CSS-Tricks. <<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>>. Luettu 15.4.2020.
- 45 A Complete Guide to Grid. Verkkoaineisto. CSS-Tricks. <<https://css-tricks.com/snippets/css/complete-guide-grid/>>. Luettu 15.4.2020.
- 46 Grid system. Verkkoaineisto. Bootstrap team. <<https://getbootstrap.com/docs/4.0/layout/grid/>>. Luettu 15.4.2020.
- 47 BuiltWith Pro. Verkkoaineisto. BuiltWith. <<https://pro.builtwith.com/newReport/technology>>. Luettu 15.4.2020.

Portfoliosivu älypuhelimien näytöllä

