Teppo Meriläinen

Julkaisujärjestelmä modulaarisen ja mukautuvan käyttöliittymän alustana

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Mediatekniikan koulutusohjelma

Insinöörityö

2.5.2016

Metropolia

Tekijä Otsikko Sivumäärä Aika	Teppo Meriläinen Julkaisujärjestelmä modulaarisen ja mukautuvan käyttöliittymän alustana 36 sivua 2.5.2016
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Mediatekniikka
Suuntautumisvaihtoehto	Digitaalinen media
Ohjaajat	Käyttöliittymäsuunnittelija Jari Vahtola Yliopettaja Kari Aaltonen

Insinöörityön tarkoituksena oli digitaalisiin ja räätälöityihin palvelukonaisuuksiin erikoistuneen yrityksen verkkosivujen tekninen suunnittelu ja toteutus. Työssä olennaista oli modulaarinen ja räätälöity käyttöliittymä sivustolla, joka rakentuu moderneista ja responsiivista tekniikoista. Sisältöä hallinnoidaan julkaisujärjestelmällä, jonka teema on yksilölisesti suunniteltu ja toteutettu erityisesti käyttötarpeiden mukaan.

Mukautuvassa web-suunnittelussa pyritään suunnittelemaan ja toteuttamaan sivustoja, jotka toimivat, tarjoavat parhaan mahdollisen käyttökokemuksen ja näyttävät hyvältä erikokoisilla laitteilla. Responsiivisuuden myötä sovelluskehyksien ja ruudukkosysteemien käyttö on yleistynyt. Rutiininomaiset tehtävät on muutettu moduuleiksi, joita voidaan kopioida ja käyttää uudestaan. Ruudukot auttavat hahmottamaan mittasuhteet ja luovat suuntaviivat sommittelulle. Yleisin tapa on käyttää 12 sarakkeen ruudukkoa.

Sivuston käyttöliittymä pohjautuu tarkkaan ruudukkoon, joka on modulaarinen ja mukautuva. Käyttöliittymä rakentuu komponenteista, joiden esitystapa ei riipu sivusta tai sivupohjasta. Esitystavat ja toiminnot standardisoitiin, jotta jokainen komponentti näkyy sivustolla oikein. Käyttöliittymä mukautuu ominaisuuksien perustella ja tarjoaa parhaan mahdollisen käyttökokemuksen laitteesta riippumatta.

Sivusto testattiin erilaisilla laitteilla, jotta voitiin arvioida sen monipuolisuus ja toimintakyky. Tuloksena saatiin eheä ja käytettävä sivusto, joka toimii mobiilissa, tabletissa ja tietokoneella.

Avainsanat	responsiivinen suunnittelu, sovelluskehykset, Wordpress

Author	Teppo Meriläinen
Title	Content management system as a base for modular and grid based responsive user interface
Number of Pages	36 pages
Date	2 May 2016
Degree	Bachelor of Engineering
Degree Programme	Media Technology
Specialisation option	Digital Media
Instructors	Jari Vahtola, UI designer Kari Aaltonen, Principal Lecturer

The purpose of the of the study was to implement a modular and grid based user interface built with modern and responsive technologies. The content is managed with a content management system made with a custom theme. The project was done for a digital design agency.

The thesis deals with subjects related to responsive design, grid systems and Wordpress.

The user interface strongly relies on a modular and responsive grid. The modular style was made with reusable components that fit into modules in the grid pattern. The page automatically responds to the user's preferences based on screen size, platform and orientation. The appearance and functional options do not vary page by page.

The website was tested with different devices to evaluate its diversity and functionality. The result is an integrated and useable website, which works on mobile, tablet and desktop devices.

Keywords	responsive design, front-end frameworks, grid, Wordpress

Sisällys

1 Johdanto

Insinöörityönä teen Exove Design -yrityksen verkkosivujen teknisen suunnittelun ja toteutuksen. Työn tavoite on toteuttaa vaatimusmäärittelyiden mukainen verkkosivusto. Sivujen tulee Exove Designin vaatimusmäärittelyiden mukaan olla mukautuva eli responsiivinen, käyttää moderneja selainpuolen web-kehitysteknologioita (hypertekstin merkintäkieltä, tyylitiedosto ja dynaamista komentosarjakieltä) ja Wordpress-julkaisujärjestelmää. Työssä itselleni oleellista oli toteuttaa yksilöllisesti räätälöity ratkaisu, syventyä erilaisten ruudukkojen käyttöön verkkosuunnittelussa ja selvittää, kuinka modulaarinen ruudukko toteutetaan julkaisujärjestelmän teeman hallinnoitavaksi. Tavoitteena on myös kehittyä selainpuolen dynaamisessa komentosarjakielessä (JavaScript) ja PHP Hypertext Preprocessor -ohjelmoijana. Työssä käytetään tekniikoita, jotka ovat hyvin ajankohtaisia web-suunnittelussa, ja se vaikuttaa myös työn aiheen valintaan.

Sivuilla pyritään edistämään Exove Designin markkinointia ja tunnettavuutta. Koska Exove Design on erikoistunut tarjoamaan asiakkailleen innovatiivisia ja räätälöityjä palvelukokonaisuuksia, on sivujen teknisessä toteutuksessa tärkeää käyttää mahdollisimman tuoreita työkaluja ja ratkaisuja.

Työssä käytetään lähdekirjallisuutta, verkkoartikkeleita ja työssäopittuja taitoja. Insinöörityöksi valitsin tämän aiheen, koska se on ajankohtainen ja minulle henkilökohtaisesti merkityksellinen työelämässä kehittymisen kannalta.

Mukautuvassa eli responsiivisessa web-suunnittelussa ajatuksena on, että sivujen käyttöliittymä toimii mahdollisimman monella eri laitteella. Sivustot optimoidaan erilaisten muutoskohtien mukaan näyttämään niissä mahdollisimman hyvältä ja tarjoamaan parhaan käyttökokemuksen.

Sivuston valmistuessa tavoite on arvioida sen toimivuutta erilaisilla alalla yleisesti käytetyillä laite- ja nopeustesteillä.

Insinöörityöraportin luvuissa 2–5 käsittelen yleisesti taustaa ja ajankohtaisia teemoja mukautuvassa web-suunnittelussa, ruudukoiden käytössä ja Wordpress-

julkaisualustassa. Luvuissa 6–7 käsittelen verkkosivujen suunnittelua ja toteutusta, jossa sovellan lukujen 2–5 teemoja käytännössä.

2 Web-suunnittelu ja trendit

Web-suunnittelu on prosessi, joka sisältää graafisen suunnittelun, käyttöliittymäsuunnittelun, käyttökokemuksen suunnittelun ja sovelluskehityksen. Verkkosivut rakentuvat ja koostuvat monesta eri taidosta ja osa-alueesta. Siksi usein työskennellään tiimeissä, jotka kattavat eri osia suunnitteluprosessista. Web-suunnittelussa ei suoranaisesti oteta kantaa sivuston tai sovelluksen taustajärjestelmään, vaan keskitytään siihen, mitä loppukäyttäjä näkee ja kokee käyttöliittymässä ja sen ulkoasussa. [1.]

Viime vuosina mobiililaitteet ovat yleistyneet ja muokanneet web-suunnittelua ja kehitystä. Yksi merkittävä tapaus oli huhtikuussa 2015, kun Google julkisti hakualgoritmiinsa uudistuksen. Se käytännössä pakottaa verkkosivustot optimoimaan sivunsa mobiililaitteille. Ilman mobiilioptimointia sivujen Google hakusijoitukset eivät nouse niin korkealle kuin mobiiliystävällisten sivujen. [2; 3.]

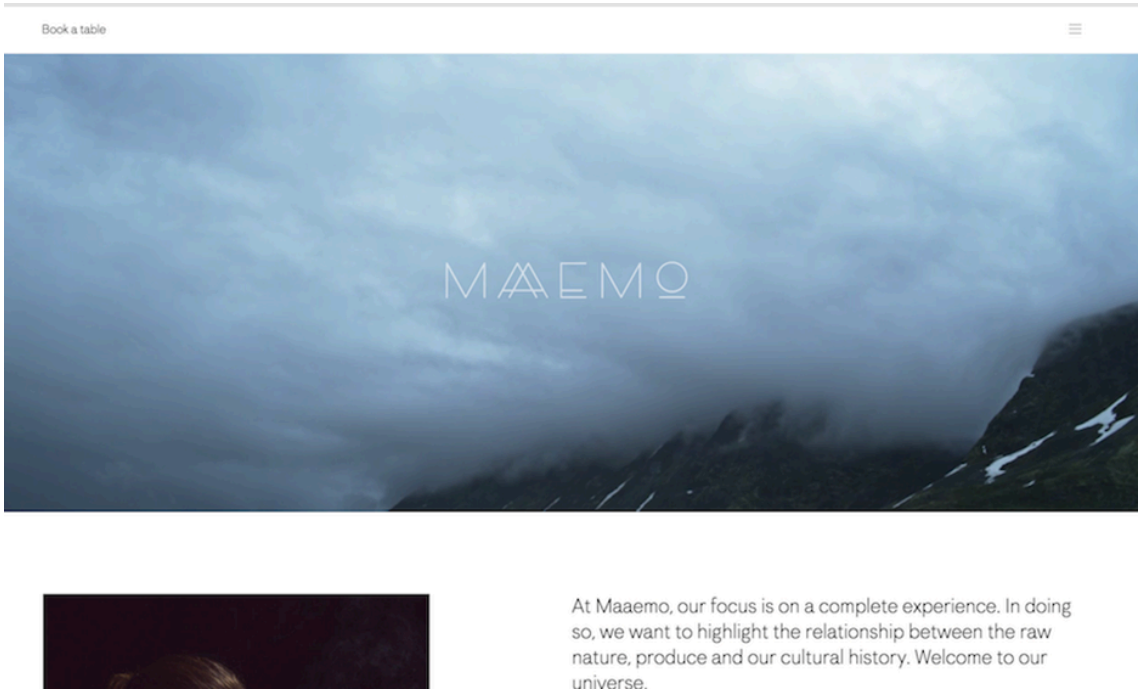
Suurin osa Internet-sivuista on nykyään mukautuvia eli responsiivisia. Sama sisältö näytetään eri laitteissa hieman eri tavalla riippuen laitteen näytön leveydestä ja korkeudesta. Sivujen pitäisi näyttää hyvältä ja olla helppokäyttöisiä riippumatta laitteesta. [2; 4.]

Vertaamalla suosituimpia sivustoja voidaan huomata selkeitä trendejä. Yksi mukautuvan web-suunnittelun sivutuotteita on, että monet sivustot muistuttavat hyvin paljon toisiaan, koska niissä on käytössä samanlaisia elementtejä, käyttöliittymäpatterneja ja komponentteja. [2.]

Selkeästi suosituin komponentti on hampurilaisvalikko eli hamburger menu. Se on navigaatioelementti, jossa on kolme viivaa allekkain. Sen visuaalinen muoto muistuttaa hieman hampurilaista. Nappia painamalla saadaan näkyviin valikko sivuista tai vaihtoehdoista. Hampurilaisvalikkoa on kritisoitu paljon, koska siinä se mikä ei ole näkyvissä, ei myöskään ole tärkeää. Oletuksena valikon sisältö on piilotettu, ja vasta nappia klikkaamalla nähdään, mitä valikko pitää sisällään. [2; 5; 6.]

Toinen malli on pitkät sivut, joita vieritetään alaspäin. Ne ovat mobiililaitteissa jo tuttu tapa ja sopivat varsinkin sivustoille, joissa käyttäjää halutaan ihastuttaa tarinankerronnalla. Sivuja rytmitetään isoilla kuvilla ja animaatioilla, joiden avulla sivuista saadaan dynaamisempia, interaktiivisia ja viihdyttäviä. [2.]

Lisäksi kuvia käytetään tavalla, jossa sivun ensinäkymässä on koko selainikkunan tai lähes koko ikkunan täyttävä niin sanottu ”hero”-kuva, joilla saadaan heti käyttäjän huomio. Teksti sijoitellaan kuvan päälle. Yleensä kuvan jälkeen seuraa siksakrakenne, jossa osiot menevät lomittain tai osiot on järjestetty ruudukkopohjaisesti laatikkoihin. Kuva 1 havainnollistaa edellä mainitut asiat. Pakkaustekniikoiden ja erilaisten JavaScriptin kirjastojen avulla kuvien kokoa ja latausaikaa on saatu mobiiliystävällisemmäksi. [2.]



Kuva 1. Esimerkki niin sanotusta hero-kuvasta [2].

Kuvien jakopalveluna ja sosiaalisenmediana tunnetun Pinterestin suosion vuoksi kortti, laatikko ja palikkamaiset elementit sekä sommitelmat ovat käytössä lähes jokaisella sivustolla, niin myös tämän opinnäytetyön sivustolla. Suorakulmion muotoiset elementit sopivat hyvin yhteen responsiivisuuden kanssa, koska niitä on helppo asetella ja koota ruudukkoon jo pelkästään niiden muodon ansiosta. [2.]

CSS-tyylitiedostojen kehittymisen myötä versiolla kolme, pelkällä tyylitiedostolla on voitu tehdä vaikuttavia animaatioita, ja sivuilla on enemmän liikettä kuin aikaisemmin. Pitkien vieritysten lisäksi animaatioita voi nähdä hiiren osoittimen hover-toiminnoissa, kuvagallerioissa ja hero-osioiden taustakuvissa. Lisäksi videot ovat yleistyneet. [2.]

Muutaman viime vuoden näkyvin trendi sivustojen visuaalisuudessa on ollut flat design. Se tarkoittaa minimalistista tyylisuuntaa, jossa käyttöliittymät tyylitellään mahdollisimman yksinkertaisesti ilman hienostelevia efektejä, jotta visuaaliseen ilmeeseen ei kiinnitettäisi liikaa huomiota. Huomio halutaan keskittää sisältöön. Minimalistinen ei tarkoita, että sivut olisivat ikävämpiä tai vähemmän graafisia kuin muut sivut. Suuret yritykset, kuten Apple ja Microsoft, ovat esimerkiksi siirtyneet käyttämään tyylisuuntaa omissa käyttöliittymissään. Sen etuna pidetään yksinkertaisemman tyylin välittämää nopeampaa viestiä. [2; 7; 8.]

3 Mukautuva web-suunnittelu

Mukautuvassa web-suunnittelussa pyritään suunnittelemaan ja toteuttamaan sivustoja, jotka reagoivat käyttäjän toimintaan ja tarjoavat parhaan mahdollisen käyttökokemuksen. Sivustojen täytyy toimia ja näyttää hyvältä erikokoisilla laitteilla. Ajatus perustuu siihen, että sivustot skaalautuvat ja mukautuvat näyttölaitteen tai selainikkunan leveyteen. Sivustot eivät reagoi laitteiden vaan ominaisuuksien perusteella. Esimerkiksi puhelimessa sisältö näytetään yhdessä sarakkeessa, kun taas tabletissa sama sisältö jaetaan kahteen sarakkeeseen. Sivujen sisältöä ja asettelua muutetaan ja mukautetaan lennosta selainikkunaan tai näyttölaitteen kokoon parhaiten soveltuvaksi. Sama sisältö siis optimoidaan täyttämään iso tai pieni ruutu mahdollisimman käyttökelpoisena ja visuaalisesti tyylikkäänä. Koska laitteiden näyttöjen koot vaihtelevat suuresti, on tärkeää, että sivusto voi mukautua mihin tahansa näytön kokoon nyt ja tulevaisuudessa [4; 9.]

Mukautuva web-suunnittelu sisältää joustavan ruudukon, joustavat kuvat ja mediakyselyt. Kaikilla näyttöko'oilla on yksi ja sama hypertext- eli HTML-tiedosto käytössä. Silloin ei välttämättä tarvita erillisiä mobiili- tai tablettisovelluksia. Riittää, että sisältö päivitetään yhteen tiedostoon. [4; 9.]

Mukautuva web-suunnittelu on myös mukautumista isompiin kokoihin. Sivut pyritään suunnittelemaan niin ikään käyttäjille, jotka selaavat sivua laitteella, jossa on hyvin iso näyttö tai resoluutio. Isommissa näyttöko'oissa kuvat voivat pikselöityä tai tekstilinjoista tulee liian pitkiä ja vaikealukuisia. [10, s. 96.]

Sisältöä pitää priorisoida käytettävyyden kannalta eri näyttöko'oissa. On tärkeää ymmärtää ja ottaa huomioon erilaiset käyttötavat ja tarpeet, riippuen siitä, käytetäänkö sivustoja älypuhelimella vai tietokoneella. [10, s. 107.] Kuva 2 esittää, miten sama sisältö on järjestetty eri näyttöko'oissa.

Kuva 2. Esimerkki mukautuvasta verkkosivusta mobiili-, tabletti- ja tietokonekoossa [4].

3.1 Joustava ruudukko

Perinteiset sivustot suunnitellaan ja toteutetaan käyttämään muuttumatonta ruudukkoa, jossa sivun leveys on kiinteä ja arvot määritellään manuaalisesti pikseleillä. Koska näyttöjen koot vaihtelevat suuresti, kiinteät ja absoluuttiset leveydet sivuissa eivät ole enää käyttökelpoinen vaihtoehto. Joustava ja skaalautuva ruudukko on parempi ratkaisu. [10, s. 28–30.]

Joustavalla ruudukolla elementit reagoivat ja skaalautuvat selainikkunan tai isäntäelementin mittasuhteisiin. Elementit saadaan muuttamaan muotoaan vaihtamalla absoluuttiset pikseliarvot prosenteiksi. Jos kiinteän sivun leveys on esimerkiksi 960 pikseliä

ja halutaan tietää 200 pikseliä leveän elementin arvo prosentteina, vastaus saadaan jakamalla 200 sivun kokonaisleveydellä 960, minkä jälkeen kerrotaan vielä saatu arvo sadalla. Lopullinen vastaus on silloin 20,83 prosenttia. Kun sivun leveys on 100 prosenttia, elementti täyttää selainikkunan leveydestä 20,83 prosenttia. Joustavassa ruudukossa sivulle määritellään enimmäisleveys, joka on yleensä 100 prosenttia. Ruudukko kannattaa myös jakaa sarakkeisiin, jotta asettelu ja elementtien hallinta pysyy selkeänä. Ruudukko ei kuitenkaan ratkaise ongelmaa, jossa useiden sarakkeiden asetteluja sovitetaan älypuhelimen näytölle. Ratkaisuksi on kehitetty CSS-mediakyselyt. [9; 10, s. 31–33.]

3.2 Mediakyselyt

Mediakyselyt keräävät dataa käyttäjästä ja hyödyntävät sitä soveltamalla CSS- eli Cascading style sheet -tyylejä, jotka voivat tarjota parhaan mahdollisen käyttökokemuksen käyttäjälle. Sisältö saadaan mukautumaan muun muassa päätelaitteen näytön resoluutioon. Kyselyt rakennetaan käyttämällä muutoskohtia, jotka perustuvat minimi- tai maksimileveysparametriin. Muutoskohtien avulla voidaan hakea tiettyjä tyylejä tyylitiedostosta selaimen ikkunan tai laitteen näytön koon perusteella. Jos näytön koko laskee tietyn koon alle, haetaan uudet tyylit edellisten sijalle. Tyylit siis tarvittaessa ylikirjoittavat aikaisemmat tyylit. CSS-tyyleillä pystytään näyttämään ja piilottamaan sisältöä, järjestämään asetteluja uudelleen ja muuttamaan kuvien kokoja. [9; 10, s. 79.]

Mediakyselyt antavat mahdollisuuden rakentaa useita asetteluja käyttämällä yhtä HTML-dokumenttia ja valikoidusti muuttamaan tyylimäärittelyt eri ominaisuuksien kuten selaimen koon, resoluution, värin tai orientoitumisen mukaan [9].

Kyselyiden muutoskohdissa voidaan käyttää minimi- tai maksimiominaisuutta. Minimiä käytettäessä tyylit toimivat vasta, kun selainikkunan leveys tai resoluutio saavuttaa muutoskohdan. Tätä tapaa käytetään esimerkiksi Mobiili ensin -konseptissa. Maksimiominaisuus taas käyttää tyylejä siihen asti, että kysely saavuttaa muutoskohdan leveyden. [10, s. 83.]

3.3 Tekstin ja kirjainten asettelu

Mukautuvassa web-suunnittelussa kirjaintyyppien eli fonttien koot merkitään em-arvona, joka määrittelee kirjaimen tarvitseman enimmäiskorkeuden. Em-arvot ovat aina suhteellisia sivun runkoon eli ”bodyyn”. Niillä on kerrannaisvaikutus, joka vaikuttaa kaikkiin asetteluihin. Jos sivun rungon leipätekstin koko on esimerkiksi 16 pikseliä ja otsikon koko 24 pikseliä, saadaan kaava 24 jaettuna 16:lla, joka on 1,5. Tällöin otsikon fonttikooksi saadaan 1,5 em. [10, s. 20.]

Jotta teksti olisi mahdollisimman luettavaa, määritellään kappaleen rivin pituus maksimileveysominaisuudella, joka on yleensä noin 66 merkkiä rivillä. Tämä voi kuitenkin vaihdella kirjaintyypistä riippuen. [11.]

3.4 Joustavat kuvat

Yksi mukautuvan web-suunnittelun suurimmista haasteista on kuvien asemointi. Siihen on kehitetty useita erilaisia ratkaisuja. Niistä yleisimmät ovat maksimileveys, CSS-overflow, mukautuvat kuvat ja PictureFill.

Maksimileveys

Kuville ei anneta leveys- eikä korkeusattribuutteja. Tyylitiedoston tyyleissä määritellään kuville maksimileveys 100 prosenttia. Niin kauan kuin mikään muu leveysperustainen kuvatyyli ei ylikirjoita tätä sääntöä, jokainen kuva ladataan alkuperäisessä koossa. Kuvan koko on 100 prosenttia selaimen koosta. Kun selaimen koko pienenee, myös kuvan koko pienenee. Kuvan täytyy silloin olla riittävän iso skaalautuakseen myös selainikkunan kasvaessa. Samaa tekniikkaa voidaan käyttää myös videoihin tai interaktiivisiin medioihin. [9; 10, s. 45, 47–49; 12.]

CSS-overflow

CSS-overflow menetelmä käyttää CSS-overflow-ominaisuutta, joka leikkaa ylimenevät osat ja sovittaa kuvan tietylle alueelle. Kuvat pysyvät alkuperäisessä koossa, mutta näkyvä osa riippuu sen alueen koosta, johon kuva on sisällytetty. [9; 10, s. 59–60.]

Mukautuvat kuvat

Kuvasta luodaan useita versioita ja ne tallennetaan palvelimelle. Sivusto valitsee sopivimman kuvan riippuen selainikkunan koosta. [12.]

PictureFill

Filament Groupin menetelmä, Picturefill pienentää kuvan resoluutiota kun se näytetään mobiililaitteessa. Näin saadaan pienennettyä latausaikoja. [9; 13; 14.]

3.5 Mobiili ensin -konsepti

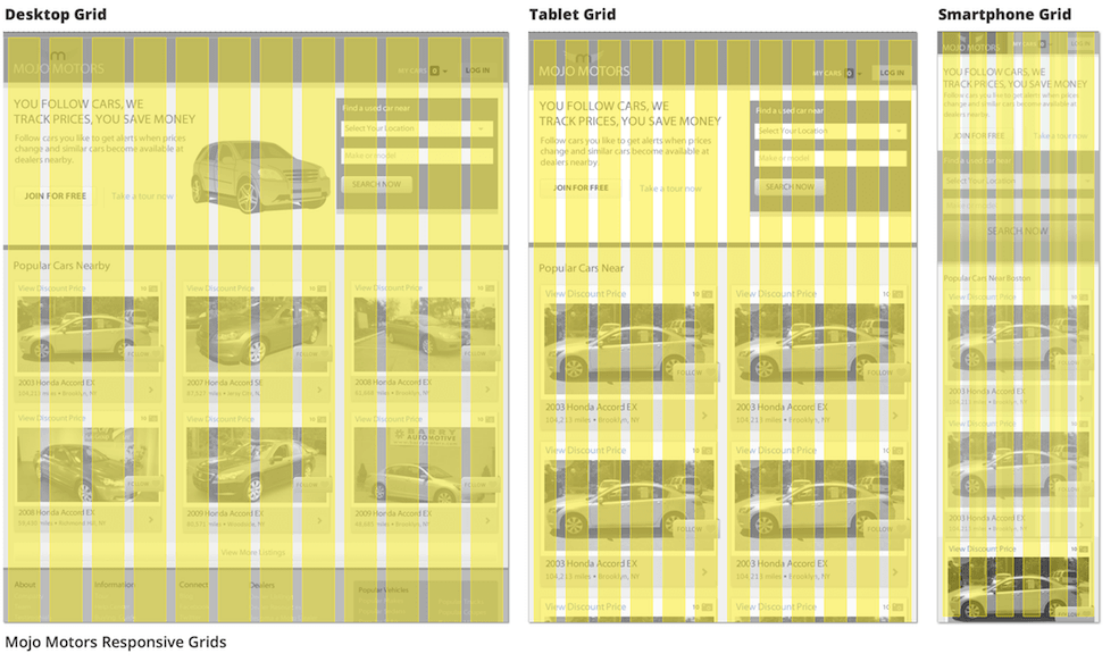
Perinteisesti verkkosivuja on suunniteltu tietokoneen ruutuun sopiviksi. Mobiiliin ei ole kiinnitetty huomiota, ja siksi monien sivujen selaaminen mobiililaitteella on ollut raskasta ja hankalaa. Viimeisen viiden vuoden aikana tilanne on kuitenkin muuttunut niin huomattavasti älypuhelinten myötä, että suunnittelussa on pakko huomioida mobiilikäyttäjät. Mobiili ensin -konseptissa suunnittelu ja toteutus aloitetaan mobiililaitteella näytettävästä koosta. Koska sisällölle ei ole niin paljon tilaa kuin tietokoneella, joudutaan rakennetta ja sisällön prioriteetteja miettimään perusteellisemmin. Mobiili ensin -konseptin ydinajatus on, että jos sisältö suunnitellaan mobiililaitetta varten tiettyyn tärkeysjärjestykseen, miksi järjestystä pitäisi muuttaa, kun on enemmän näyttötilaa. Mobiililaitteen näytölle suunniteltaessa on suunnittelussa pakko keskittyä kriittisiin ja olennaisiin näkökohtiin. Mobiili on rajoitetuin koko, missä suunnittelijat ja kehittäjät työskentelevät. Sisältöä asetellaan progressiivisesti mediakyselyiden avulla. Lisäominaisuuksia voidaan helposti lisätä, kun koot muuttuvat suuremmiksi. Kaikkea sisältöä ei välttämättä siis tarvitse näyttää jo mobiilikoossa. Tämä myös tarjoaa parempaa tukea mediakyselyihin ja muihin optioihin. [10, s. 111–112; 15, s. 1.]

4 Ruudukkojen käyttö web-suunnittelussa

4.1 Ruudukkoon pohjautuva web-suunnittelu

Suurin osa tämänhetkisistä verkkosivuista pohjautuu ruudukkoon ja ruudukkojärjestelmiin. Ruudukkoja on käytetty jo pitkään printti- ja graafisessa suunnittelussa, mutta

vasta mukautuvan web-suunnittelun yleistyessä ruudukoista ja tyylitiedoston ruudukkoihin pohjautuvista sovelluskehyksistä on tullut standardi verkkomaailmassa. Ruudukkoon pohjautuvassa web-suunnittelussa sivu jaetaan tasaisiin kolumneihin. Ruudukot rakentuvat kolumneista ja guttereista eli suoja-alueista kolumnien välissä. [16; 17; 18.] Kuva 3 havainnollistaa ruudukon kolumnit ja suoja-alueet.



Kuva 3. Smashing magazinen esimerkki mukautuvasta ruudukosta [19].

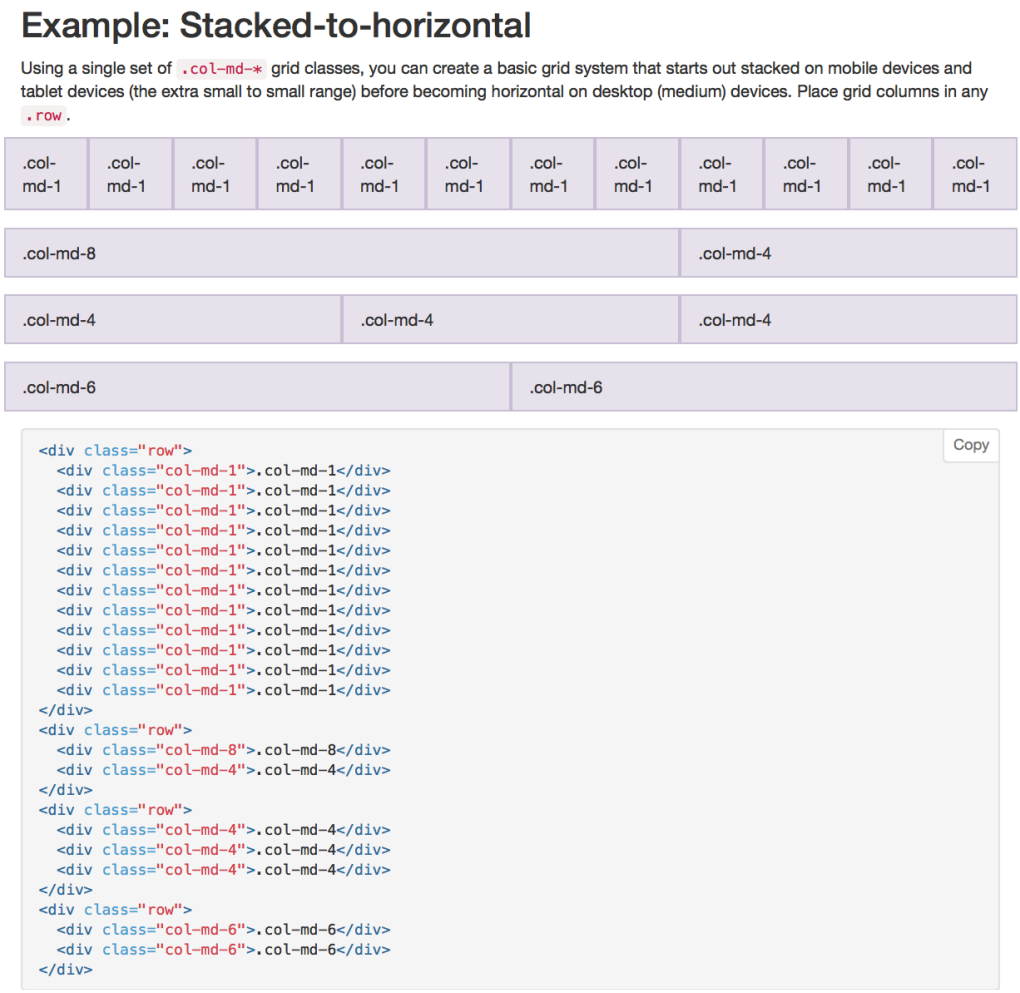
Keskeisintä ruudukoissa on, että ne auttavat hahmottamaan mittasuhteet ja luovat suuntaviivat sommittelulle. Ruudukoiden avulla suunnittelussa säilyy tasapaino ja rytmi. Ne luovat perustan yhtenäiselle ja esteettisesti miellyttävälle taitolle. Yksi ruudukon suurimmista eduista on, että sitä on helppo käyttää verkkosivun taiton suunnittelussa Photoshop-ohjelmassa ja sama ruudukko on käytettävissä selainpuolen sovelluskehyksessä. Ruudukon käyttö helpottaa kommunikointia graafisten suunnittelijoiden ja käyttöliittymäkehittäjien välillä. Yleisin tapa on käyttää 12 kolumnin ruudukkoa, jolloin yhden kolumnin leveys on 1/12 sivun leveydestä. Suurin osa selainpuolen sovelluskehyksistä ja tyylikirjastoista käyttää 12:lla jaollista ruudukkoa. [16; 17; 18; 20.]

4.2 Sovelluskehykset

Sovelluskehys on joukko uudelleenkäytettäviä työkaluja, kirjastoja, tapoja ja parhaita käytäntöjä. Rutiininomaiset tehtävät on muutettu moduuleiksi, joita voidaan kopioida ja käyttää uudestaan. Tällöin suunnittelija tai kehittäjä pystyy paremmin keskittymään yksilöllisiin ja räätälöityihin tehtäviin. Se on alusta, jonka avulla verkkosivuja koostetaan toiminnallisiksi käyttöliittymiksi. [21.]

Sovelluskehyksien avulla pyritään valmistamaan rakenteellisia ja uudelleenkäytettäviä komponentteja, joista web-käyttöliittymä voidaan rakentaa. Selainpuolen sovelluskehykset rakentuvat kolmesta osasta: Cascaing style sheet -tyylitiedostoista, hyperteksti merkintäkielikomponenteista ja dynaamisista ja toiminnallisista JavaScript- eli selainpuolen komentosarjakielen komponenteista. Tyylitiedostot luovat ruudukon ja sivuston ilmeen. Yleensä ilme on hyvin minimalistinen, jotta tyylejä pystytään helposti muokkaamaan. Hypertekstikomponentit sisältävät muun muassa lomakkeet, napit, navigaation ja alasvetovalikot. Toiminnallisia JavaScript-komponentteja ovat esimerkiksi karusellit ja modaalit. [22.]

Suosituimmat selainpuolen sovelluskehykset verkossa ovat tällä hetkellä Bootstrap (kuva 4) ja Foundation. Järjestelmät sisältävät luokkia, jolla sivuston elementtejä pystytään jakamaan pysty- ja vaakasuunnassa osioihin. [22; 23.]



Kuva 4. Bootstrapin ruudukot ja luokat [25].

Bootstrap on ilmainen avoimen lähdekoodin sovelluskehys, jolla voi luoda verkkosivuja ja web-sovelluksia. Se on tämän hetken suosituin selainpuolen sovelluskehys. Sen suosio ei johdu sen teknisestä paremmuudesta vaan sen levinneisyydestä Internetissä. Miljoonat verkkosivut on toteutettu Bootstrapillä. Googlen haulla löytää lukuisia artikkeleita ja oppaita, jossa syvennytään siihen tarkemmin. Lisäksi se on selkeästi ja tarkasti dokumentoitu omilla verkkosivuillaan. Kuka vain pystyy halutessaan oppimaan käyttämään sitä omissa projekteissaan. [23; 24.]

Bootstrapissa row-luokka luo vaakasuuntaisia rivejä joiden sisälle kolumnit asetellaan. Näitä sarakkeita kutsutaan col-luokiksi. Kuvassa 4 on esimerkki kuinka col-luokkia voidaan käyttää Bootstrapissä. Mukautuvuus rakennetaan sarakkeen perään annettavilla luokilla. Jos esimerkiksi halutaan, että mobiilinäkymässä osio on puolet ruudun leveydestä annetaan sille luokka col-xs-6. Bootstrapissa on 12:lla jaollinen ruudukkosystee-

mi, joten logiikka toimii tässä tapauksessa 12/2 = 6. Muut näyttökoot ovat sm, md ja lg. Sm vastaa tablettikokoa, md taas tietokonetta ja lg erittäin suuria näyttöjä. Numeroilla kerrotaan, kuinka monta saraketta leveä osio on. Luokkien avulla voidaan hyvin tarkasti määritellä, minkä levyisiä osioita ja elementtejä sivu pitää sisällään eri selainikkunanleveyksissä. Yhteen osioon tai elementtiin voidaan myös laittaa useampi luokka. Esimerkiksi jos osio on mobiilissa koko ruudun levyinen eli 12 saraketta, mutta tabletissa 6 ja tietokoneella vain 4 saraketta, tällöin täytyy osiolle antaa luokat col-xs-12, col-sm-6 ja col-md-4. Luokat käyttävät hyväkseen Bootstrapiin rakennettuja mediakyselyitä, joista on esimerkki kuvassa 5. [25.]

Kuva 5. Bootstrapin mediakyselyt [25].

Bootstrap tarjoaa kehikon, jonka päälle sivusto voidaan rakentaa. Sen sisältämät komponentit ja luokat löytyvät sen omilta verkkosivuilta, joita voidaan soveltaa omiin käyttötarpeisiin. Itse pidän Bootstrapissä eniten sen ruudukkosysteemistä. Lisäksi siinä on lukuisia valmiita JavaScript-komponentteja, joilla sivuille saadaan vaivattomasti dynaamisuutta, kuten modaaleita, tooltippejä tai alasvetovalikoita.

Sovelluskehyksien hyötyjä ovat muun muassa mukautuvuus, siisti ja hyvin jäsennelty koodi, yhteensopivuus selaimien kanssa, nopea kehittäminen ja valmis ruudukkosysteemi. Sovelluskehykset myös opettavat ajattelemaan ja suunnittelemaan komponentteihin ja ruudukkoihin pohjautuen. [22; 26; 27.]

Sovelluskehyksissä on myös heikkouksia. Niitä ovat muun muassa turha koodi, komponentit joille ei ole käyttöä, ja tarve opetella jokaisen sovelluskehyksen tavat. Joissain tapauksissa koodin kirjoittaminen itse alusta alkaen voi olla järkevämpää ja jopa nopeampaa. Jos web-suunnittelijalla on kokemusta vain sovelluskehyksien valmiista koodis-

ta, voi myös olla vaikeaa oppia käyttämään selainpuolen teknologioita oikein. [22; 26; 27.]

Sovelluskehykset ovat vakiinnuttaneet asemansa web-suunnittelussa, ja niistä näyttäisi olevan enemmän hyötyä kuin haittaa. Ne nopeuttavat web-kehittämistä ja -suunnittelua, varsinkin kun puhutaan ruudukoista ja ruudukkopohjaisista käyttöliittymistä. Sivuston runko ja asettelu on mahdollista koota nopeasti, varsinkin jos sovelluskehyksen komponentit ja luokat ovat jo ennestään tuttuja.

5 Wordpress-julkaisujärjestelmä

Wordpress on verkko-sovellus, jolla pystytään luomaan ja hallitsemaan verkkosivuja, blogeja tai sovelluksia. Toisin sanoen se on julkaisujärjestelmä, jolla voidaan luoda ja päivittää sisältöä verkkosivulle ilman, että käyttäjällä on kovin laajaa, tai minkäänlaista teknistä osaamista hypertekstin merkintä- tai PHP Hypertext Preprocessor -kielistä. Se on ilmainen avoimen lähdekoodin julkaisualusta, joka on kirjoitettu PHP Hypertext Preprocessor -kielellä ja tietojen tallentamiseen käytetään MySql- eli relaatiotietokanta-ohljelmistotietokantaa. Wordpress on Internetin suosituin sisällönhallintaohjelmisto, ja sitä on kehitetty vuodesta 2003 lähtien. Sillä on toteutettu yli 60 miljoona verkkosivua. Alusta koostuu teemasta, lisäosista ja pienoisohjelmista, joita Wordpress nimittää vimpaimiksi. [28; 29.]

5.1 Teemat

Teemojen avulla käyttäjät voivat muuttaa sivuston ulkoasua ja toiminnallisuutta ilman, että ydinjärjestelmän koodi muuttuu. Jokainen Wordpress-sivusto vaatii ainakin yhden standardin mukaisen teeman toimiakseen. Teeman täytyy sisältää seuraavat kolme asiaa: jäsenneltyä PHP Hypertext Preprocessor -kieltä. Hypertext-merkintäkieltä ja Cascading style sheets -tyylitiedoston. PHP-kielellä muodostetaan mallitiedostot, jotka ohjaavat, miten sivusto tuottaa informaatiota näkyviin tietokannasta. Hypertext-merkintäkieli luo sivuston elementit, ja sen täytyy olla semanttista ja validia. CSS-tyyliedostot luovat visuaaliseen ilmeen. Nämä tiedostot rakentavat yhdessä graafisen käyttöliittymän. Lisäksi teemat sisältävät kuvia ja JavaScript-tiedostoja. [30.]

Alustaan on saatavilla lukuisia teemoja, joita käyttäjät voivat itse asentaa. On olemassa sekä ilmaisia että maksullisia teemoja. Maksulliset teemat yleensä sisältävät visuaalisesti hienoja komponentteja ja efektejä. Valmiit teemat sopivat parhaiten niille, jotka haluavat saada verkkosivun nopeasti ja vaivattomasti valmiiksi. Mahdollista on myös rakentaa kokonaan oma teema. Wordpress on erittäin hyvin dokumentoitu ja ohjeistettu, ja sillä on myös aktiivinen käyttäjäkunta ympäri maailman. Valmiitakin teemoja pystyy räätälöimään käyttämällä lapsi- eli child-theme-ominaisuutta. Lapsi-teema on teema, joka perii toiminnallisuuden ja muotoilun toisesta teemasta, jota kutsutaan isäntä eli parent-teemaksi. Lapsi-teemoja suositellaan käytettävän silloin, kun halutaan muokata teemaa. [30; 31.]

5.2 Lisäosat

Wordpressiin voidaan asentaa lisäosia, joiden avulla verkkosivulle saadaan lisää ominaisuuksia. Lisäosa on ohjelma tai sarja toimintoja, jotka on ohjelmoitu PHP Hypertext Preprocessor -kielellä. Lisäosat toimivat saumattomasti Wordpressin kanssa, ja niiden asentaminen ja poistaminen on helppoa eikä teknistä osaamista yleensä tarvita. Lisäosa voidaan asentaa hallintapaneelista haulla tai lataamalla tiedosto järjestelmään suoraan. Haku löytää pelkästään lisäosat, jotka on rekisteröity Wordpress lisäosahakemistoon. Sen jälkeen lisäosa aktivoidaan nappia painamalla. Joissain tapauksissa yhteensopivuusongelmia voi ilmetä, jos sivustolla on samanaikaisesti käytössä monia liitännäisiä. Suositumpia lisäosia ovat kokonaiset verkkokaupat, erilaiset kuvagalleriat, hakukoneoptimointi, lomakkeet ja hallintatyökalut. Lisäosia on saatavilla yli 40 000, ja ne kukin tarjoavat räätälöityjä toimintoja ja ominaisuuksia käyttäjän erilaisiin tarpeisiin. Lisäosia käytetään silloin, kuin teemasta ei löydy ratkaisua haluttuun ominaisuuteen tai toiminnallisuuteen. Äärimmäisessä tapauksessa kyseessä voi olla verkkokauppalisäosa, joka mahdollistaa toimivan ja hallitavan verkkokaupan. Kuten teemoja, myös lisäosia on sekä maksullisia että ilmaisia. Yleensä maksulliset liitännäiset tarjoavat monipuolisempia ominaisuuksia ja parempaa teknistä tukea mahdollisissa vioissa ja ongelmatilanteissa. Varsinkin ilmaisten liitännäisten kanssa tulee olla hyvin tarkka päivitysten suhteen. Kaikkia liitännäisiä ei päivitetä aktiivisesti, josta voi seurata tietoturvauhkia ja toiminnallisia vikoja. Mitä enemmän liitännäisiä sivuilla on käytössä, sitä hitaammat ne myös ovat. [32; 33; 34.]

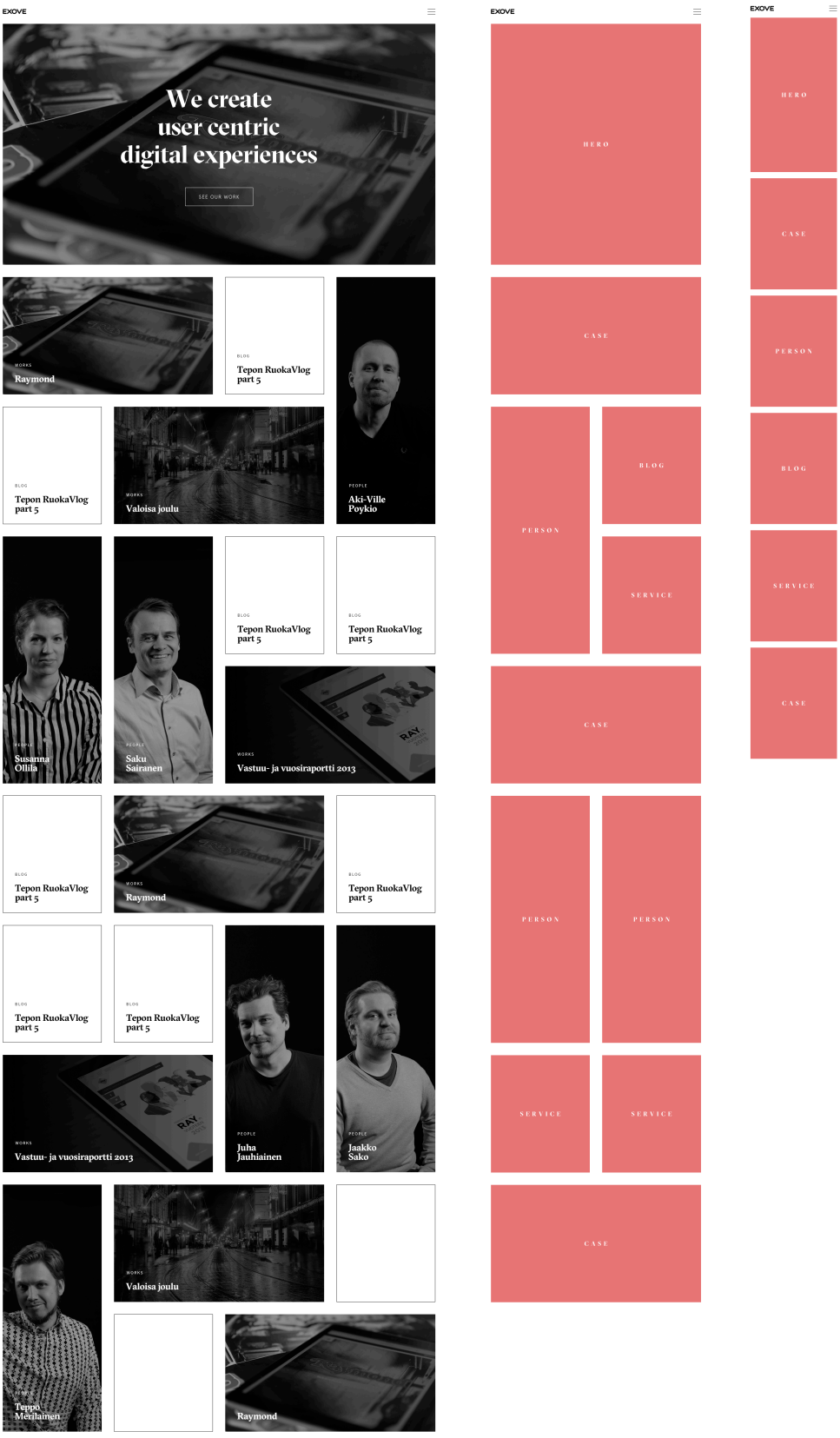
6 Verkkosivun suunnittelu

Tässä luvussa käsittelen insinöörityön suunnittelua. Suunnittelin ja toteutin verkkosivut Exove Design-yritykselle. Se on suunnittelutoimisto, joka on erikoistunut digitaalisiin palvelukokonaisuuksiin. Verkkosivujen tarkoitus on esitellä yritystä ja sen työntekijöitä asiakkaille. Sivuille haluttiin myös Blog-osio, johon voidaan päivittää ajankohtaista sisältöä.

6.1 Sivupohjat ja tyylittelyt

Suunniteltu sivusto rakentuu kolmesta sivupohjasta: etu-, kategoria- ja artikkelisivusta. Artikkeleilla on kolme pääkategoriaa: blog, works ja people. Etusivu listaa kaiken sisällön, kun taas kategoriasivu näyttää vain yhtä kategoriaa kerrallaan, määritellystä kategoriasta riippuen. Artikkelisivu näyttää koko artikkelin sisällön. Listaavissa sivuissa sisältö tyylitellään kategorian luokan mukaan. Luokat tyylitellään ja näytetään erimuotoisena listaavilla sivulla, ja ne suhteutetaan toisiinsa visuaalisesti. Peruselementti on works-luokka, jonka kokosuhde on noin 16/9 eli geometriseltä muodoltaan vaakasuorakolmio. Blog-luokat ovat yhtä korkeita ja leveyssuunnassa puolet pienempiä elementtejä kuin works-luokat. People-luokat taas ovat yhtä leveitä kuin blog-luokat, mutta kaksinkertaisesti korkeammat kuin blog- tai works-luokat.

Etusivu sisältää hero-osion ja ruudukon. Kuva 6 osoittaa, kuinka ruudukkoon järjestellään sisältöä peräkkäin kategorioiden tyylien mukaisesti.



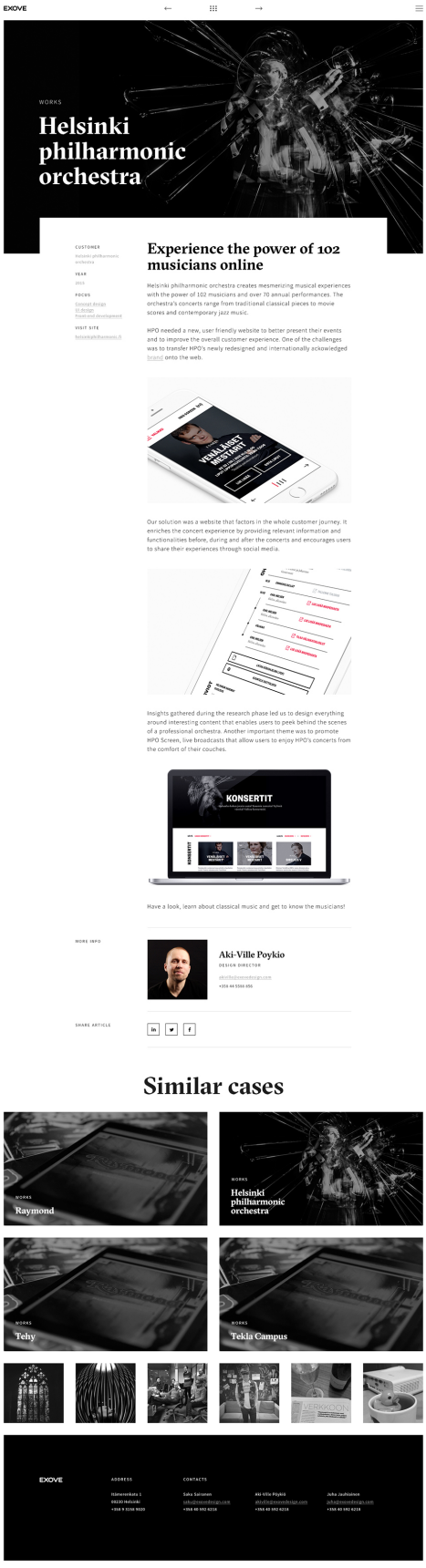
Kuva 6. Suunnitellun sivuston etusivun rakenne.

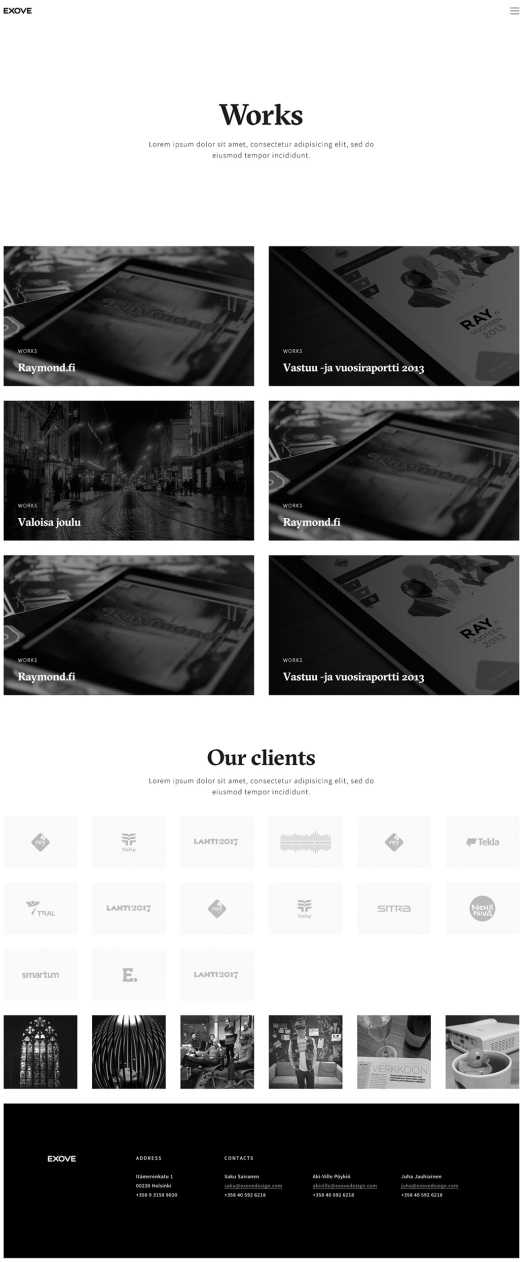
Toistuvia modulaarisia komponentteja sivuston ilmeessä ovat hero-osiot, blog-, works- ja people-komponentit. Niiden ulkonäkö on ennalta määrätty eikä vaihtele. Blog-, works- ja people-komponenttien rakenne on hyvin samankaltainen toisiinsa nähden. Niissä kategoria ja otsikko sijoitellaan vasempaan alareunaan.

Artikkelisivu koostuvat hero-osiosta, sivun sisällöstä ja aiheeseen liittyvästä lohkosta. Sisältöalueessa on kaksi osiota: sivupalkki ja runko. Sivupalkkiin sijoitellaan artikkelin metadataa ja runko sisältää tekstiä ja kuvia. Aiheeseen liittyviä aiheita nostetaan ruudukkoon.

Kategoriasivut rakentuvat hero-osiosta ja ruudukosta.

Kuvassa 7, vasemmalla puolella on esimerkki artikkelisivun suunnitellusta rakenteesta ja oikealla kategoriasivun rakenne.

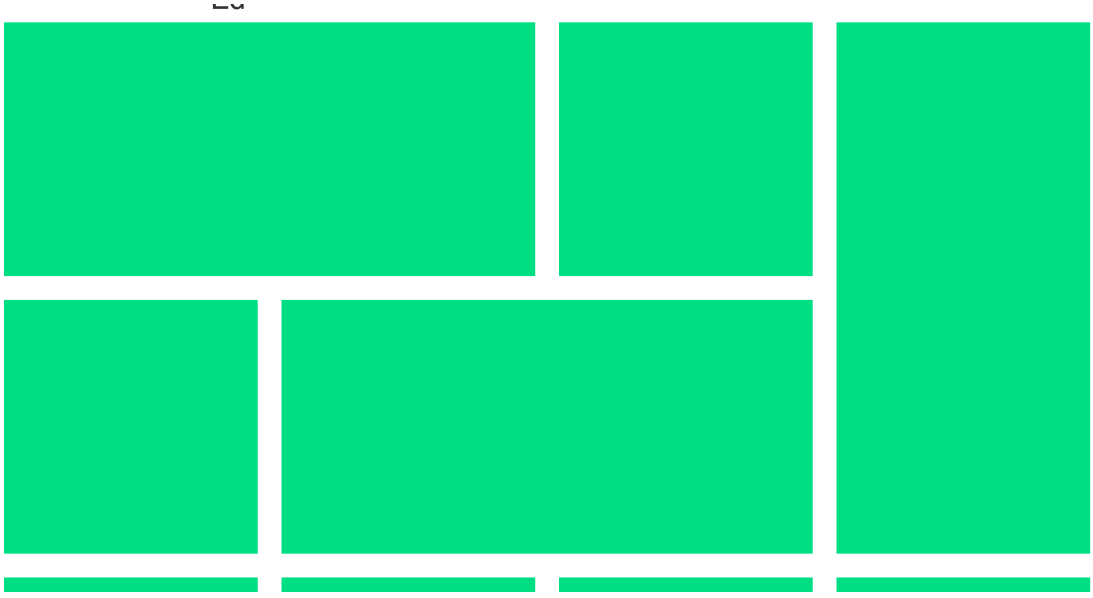




Kuva 7. Suunnitellun sivuston artikkeli- ja kategoriasivun rakenne.

Listaavilla sivuilla ruudukko on avainasemassa. Se on myös teknisesti projektin haastavin osuus. Ruudukosta täytyy saada aukoton, eli sisällön täytyy kellua ja täyttää tyhjät kohdat ruudukossa. Ensimmäisenä lähdin ratkaisemaan, voiko ruudukon toteuttaa järkevästi pelkästään Cascading style sheets tyyleillä. Huomasin, että ruudukon pystyisi luomaan pelkällä tyylitiedostolla, jos elementtien järjestys olisi aina sama. Tämä ei kuitenkaan riittänyt ratkaisuksi, koska järjestystä täytyi tarvittaessa pystyä muokkaamaan. Siirryin JavaScriptin kirjastojen pariin. Pian löysin Isotope masonry -kirjaston, joka asettelee elementtejä optimaaliseen sijaintiin riippuen kuinka paljon elementille on tilaa pystysuunnassa. Isotopen kirjastolla ei kuitenkaan saanut aikaiseksi täysin aukottomia ruudukkoja. Lopulta päädyin Packery-kirjastoon. [35.]

Testasin ensin Packery-kirjastoa prototyypillä, jota Kuva 8 esittää. Kuvat 8, 9 ja 10 havainnollistavat prototyyppiin toiminnallisuuteen vaikuttavat tiedostot.



Kuva 8. Suunnitellun sivuston prototyypin käyttöliittymä.

```
<div >
    <div id="grid">
      <div class="grid-item grid-item--width2 work"></div>
      <div class="grid-item blog"></div>
      <div class="grid-item grid-item--height2 person"></div>
      <div class="grid-item blog"></div>
      <div class="grid-item grid-item--width2 work"></div>
      <div class="grid-item grid-item--height2 person"></div>
      <div class="grid-item grid-item--height2 person"></div>
      <div class="grid-item blog"></div>
      <div class="grid-item blog"></div>
      <div class="grid-item grid-item--width2 work"></div>
      <div class="grid-item blog"></div>
      <div class="grid-item grid-item--width2 work"></div>
      <div class="grid-item blog"></div>
      <div class="grid-item blog"></div>
      <div class="grid-item blog"></div>
      <div class="grid-item grid-item--height2 person"></div>
      <div class="grid-item grid-item--height2 person"></div>
      <div class="grid-item grid-item--width2 work"></div>
      <div class="grid-item grid-item--height2 person"></div>
      <div class="grid-item blog"></div>
      <div class="grid-item grid-item--width2 work"></div>
      <div class="grid-item blog"></div>
      <div class="grid-item grid-item--width2 work"></div>
      <div class="grid-item blog"></div>
      <div class="grid-item blog"></div>
      <div class="grid-item blog"></div>
      <div class="grid-item blog"></div>
    </div>
</div>
```

Kuva 9. Suunnitellun sivuston prototyypin HTML-merkintäkieltä.

```
/* ---- .grid-item ---- */
.grid-item {
  float: left;
  height: 320px;
  background: #0D8;
  width: 320px;
}
@media (min-width: 670px) {
  .grid-item--width2 {
    width: 670px;
  }
}
@media (min-width: 670px) {
  .grid-item--height2 {
    height: 670px;
  }
}
```

Kuva 10. Suunnitellun sivuston prototyypin ruudukon tyylitiedosto.

```
$(function() {
    var $container = $('#grid');
    // init
    $container.packery({
      itemSelector: '.grid-item',
      gutter: 30
    });
});
```

Kuva 11. Suunnittelun sivuston prototyypin JavaScript-tiedosto.

Aluksi muodostin elementit ja suoja-alueet pikseleitä käyttäen. Elementeistä pikseleillä ei kuitenkaan saanut tarpeeksi mukautuvia, koska ne eivät skaalautuneet eivätkä sopeutuneet tarpeeksi joustavasti selainikkunan muutoksiin. Ratkaisuna ongelmaan päätin, että sivusto pyrkii käyttämään prosentteja pikseleiden sijaan, jotta sivustosta saataisiin mahdollisimman adaptiivinen ja responsiivinen.

Artikkelisivuilla päätin käyttää Bootstrapin ruudukkosysteemiä, koska se oli jo ennestään tuttu ja hyväksi havaittu.

6.2 Kuvat ja videot

Kuvilla ja videoilla on merkittävä rooli sivun visuaalisuudessa. Listaavilla sivuilla, joissa on käytössä Packery-ruudukko, kuvat ja videot näytetään aluksi mustavalkoisina. Kun hiiren osoitin osoitetaan osioon jossa on kuva tai video, ne muuttuvat värilliseksi. Tätä kutsutaan hover-toiminnoksi. Jos kyseessä on video, se alkaa värien muuttumisen yhteydessä toistua. Hiiren osoittimen poistuessa videon tai kuvan päältä toiminnot palaavat ennalleen. Jos videoon osoitetaan uudestaan, se alkaa toistumaan samasta kohdasta mihin se edellisellä kerralla jäi. Otsakkeen kuvat tai videot ovat aina selainikkunan korkuisia. Niissä ei ole käytössä samanlaista hover-toimintoa vaan ne ovat aina värillisiä. Artikkelisivujen sisältöosiossa kuvat ovat myös aina värillisiä.

6.3 Mukautuvuus

Yksi sivuston tärkeimmistä piirteistä visuaalisuuden ja ruudukon lisäksi on mukautuvuus. Osioiden ja niiden sisällön täytyy näyttää visuaalisesti tasapainoiselta ja toimia eri päälaiteissa kuten älypuhelimissa, tableteissa ja tietokoneilla. Jotta tämä onnistuisi mahdollisimman hyvin, päätin soveltaa mobiili ensin -konseptia sivustolle, eli sen to-

teuttaminen aloitettiin mobiilinäkymästä. Tätä tukee myös se, että Bootstrap, joka on osittain käytössä sivustolla, on mobiiliystävällinen ja perustuu mobiili ensin -konseptiin. Suurin haaste sivuston mukautuvuudessa ovat kuvat ja videot, varsinkin videot, sillä esimerkiksi iPhone ja iPad eivät tue automaattista videon toistoa. Lisäksi videot toistetaan laitteiden omassa videosovelluksessa. [25.]

Packery-ruudukkossa elementit ovat aluksi neliönmuotoisia. Vasta tablettikoossa elementit ottavat oman muotonsa ja mittasuhteet käyttöön suunniteltujen kategorialuokkien mukaisesti.

6.4 Teeman valinta

Vaihtoehtoina oli joko valita valmis teema tai käyttää aloitusteemaa. Molemmista vaihtoehdoista minulla oli aikaisempaa kokemusta. Valmis teema yleensä sopii silloin, kuin sivusto on suunniteltu valmiin teeman ohjeistuksen ja komponenttien mukaan. Tässä tapauksessa sivuston visuaalisuus ja elementit olivat kuitenkin niin yksilöllisiä, että päätin valita aloitusteeman, jotta välttyisin turhalta työltä ja koodin yliajamiselta. Minulle tutuin aloitusteema oli Rootsin kehittämä Sage. Se sisältää modernit webkehittäjän työvälineet ja työnkulun, ja siinä on osittain käytössä Bootstrap.

7 Verkkosivun toteutus ja tulokset

Tässä luvussa käsitellään verkkosivuston toteutuksen eri työvaiheita.

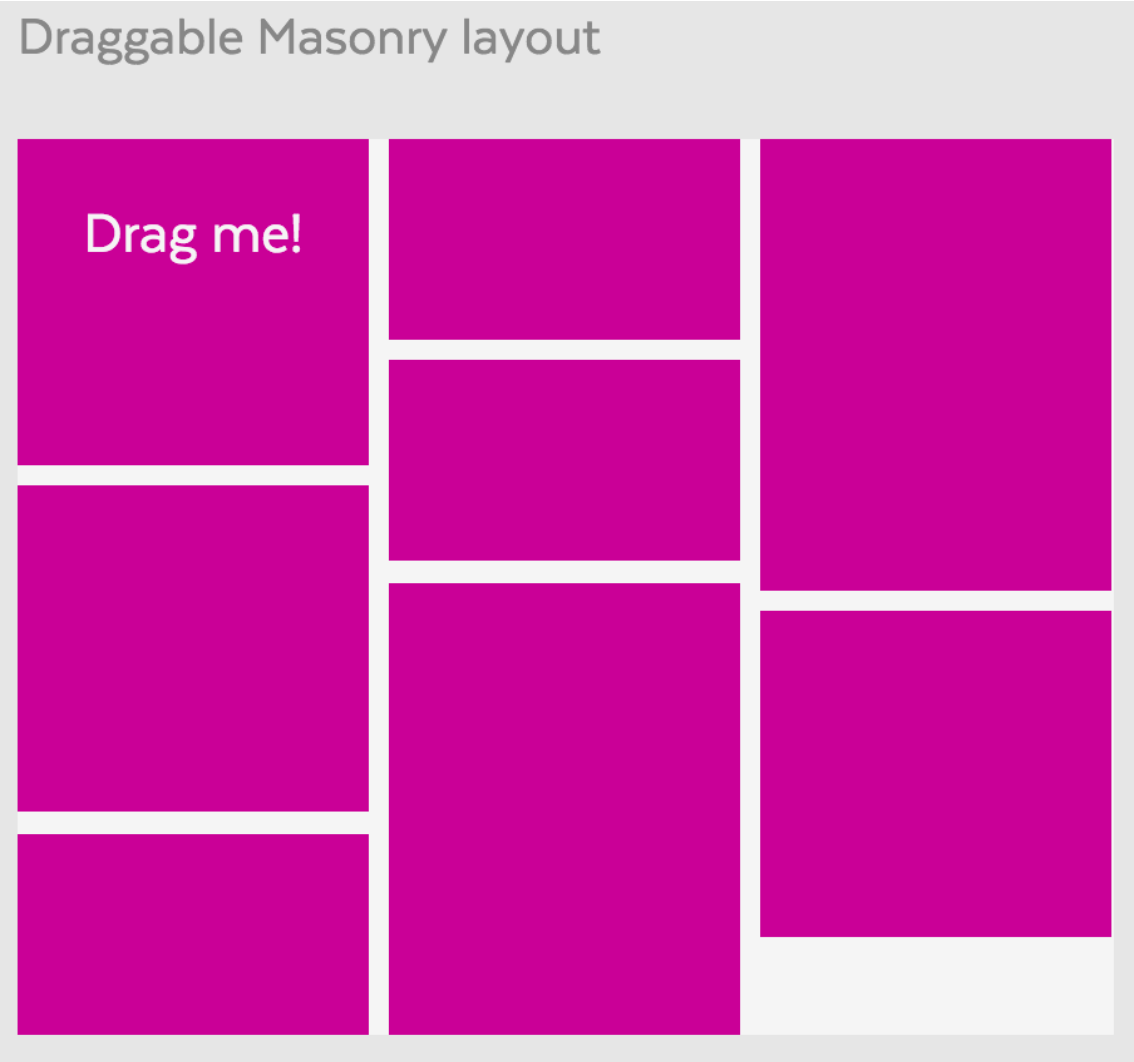
7.1 Ruudukon luominen

Artikkelisivulla sisältöalue jaetaan osiin käyttäen Bootstrapin ruudukkojärjestelmää. Koska Bootstrapin ruudukko jakautuu 12 osaan, sivun sisältöalue on kymmenen kolumnin levyinen. Sen sisällä on kaksi osiota: sivupalkki eli aside, ja runko eli body. Sivupalkki on neljän sarakkeen levyinen, joista yhden sarakkeen verran on tyhjää, jotta vasemmalle reunalla saadaan yhden sarakkeen verran tyhjää tilaa. Runko on seitsemän sarakkeen levyinen. Silloin sille jää yhden sarakkeen verran tyhjää oikealle. Mobiilissa osiot listautuvat jonoon allekkain ja ovat 12 saraketta leveitä. Lisäksi sivustolla on käytössä Bootstrapin mediakyselyt, joita hyödynnetään myös Packery-ruudukon mu-

kautuvuudessa. Kuvassa 12 nähdään, kuinka Bootstrapin luokkia on käytetty Artikkelisivulla.

Kuva 12. Artikkelisivun HTML-rakenne.

Listaavilla sivuilla ruudukossa on käytössä Packery.js, joka on Javascript-kirjasto ja Jquery-liitännäinen. Sillä saadaan aukottomia ja raahattavia Masonry-gallerioita. Masonry on tekniikka, jossa elementtejä sijoitetaan optimaalisesti ruudukkoon vierekkäin siitä riippuen, kuinka paljon elementille on tilaa ruudukon pystysuunnassa. Näin elementit saadaan tasattua aukottomasti ruudukkoon pystysuunnassa. [36; 37.] Kuva 13 esittelee masonry-gallerian elementtien sijoittelua.



Kuva 13. Masonry-galleria [37].

Olennaista ruudukon toteuttamisessa oli tyylitellä kategorialuokat suunnitelman mukaisesti (6.1 Sivupohjat ja tyylittely). Käytännössä jokaiselle grid-item-elementille lasketaan Jqueryllä korkeus, joka on suhteessa tyhjään tai piilotettuun grid-item-ratio-elementtiin Luokalla grid-item-ratio määritellään kokosuhteen 16/9 leveys ruudukossa. Se mukautuu siitä riippuen kuinka leveä selainikkuna on. Mobiilinäkymässä kokosuhde muutetaan neliöksi. Sen jälkeen tarkastetaan, mikä kategoria-luokka grid-item-elementillä on, minkä jälkeen tyylitiedosto antaa elementille leveyden ja Jquery laskee ja käsittelee korkeuden. Jotta ruudukosta saadaan aukoton ja tasainen, täytyy luokan category-people-elementtien korkeuksiin ottaa huomioon suoja-alueen koko. Korkeus näissä tapauksissa on (suhdekorkeus x 2) + suoja-alueen korkeus. Koska suurin osa sivuston pisteistä on määritelty prosentteina täytyy selaimen tallentaa suoja-alueen korkeutta muuttujaan. Kun selainikkunan leveyttä muutetaan, muuttajan arvo tarkiste-

taan ja määritellään uudestaan. JavaScript-tiedostoon tehdään funktio, joka manipuloi grid-item-luokan elementtejä ensin luomalla muuttujan suoja-alueen ja suhde-elementin korkeudesta, minkä jälkeen arvot asetetaan elementeille. Kuva 14 ja 15 havannollistaa, kuinka grid-item-ratio tyylitellään ja sijoitetaan ruudukkoon.

```
.grid-item-ratio {
  display: none;
  position: relative;
  width: 100%;
  @media (min-width: $screen-xs-min) {
    width: 100%;
  }
  @media (min-width: $screen-sm-min) {
    width: 66%;
  }
  @media (min-width: $screen-lg-min) {
    width: 49%;
  }
  @media (min-width: $screen-wlg-min) {
    width: 38.8%;
  }
  &:before {
    content: "";
    float: left;
    padding-bottom: 100%;
    @media (min-width: $screen-xs-min) {
      padding-bottom: 54%;
    }
  }
  &:after {
    content: "";
    display: table;
    clear: both;
  }
}
```

Kuva 14. Grid item ratio -luokan CSS-tyylit.

```
▼<section id="grid" style="position: relative; height: 5256.63px;">
    <div class="grid-sizer"></div>
    <div class="gutter-sizer"></div>
    <div class="grid-item-ratio"></div>
  ▼<article class="grid-item post-691 post type-post status-publish format-standard
  has-post-thumbnail hentry category-apple category-blog" style="height: 379px;
  position: absolute; left: 0px; top: 0px;">
      ::before
      <a class="overlay" href="http://exovedesign.com/apple-launches-new-apple-watch-
      page/"></a>
    ▼<div class="caption" style="position: absolute;">
        <p>Blog</p>
      ▼<h3 class="entry-title">
          <a href="http://exovedesign.com/apple-launches-new-apple-watch-page/">Apple
          launches redesigned Watch page</a>
        </h3>
      </div>
      ::after
    </article>
  ▶<article class="grid-item post-123 post type-post status-publish format-standard
```

Kuva 15. Sivuston ruudukon hypertext-merkintäkieltä.

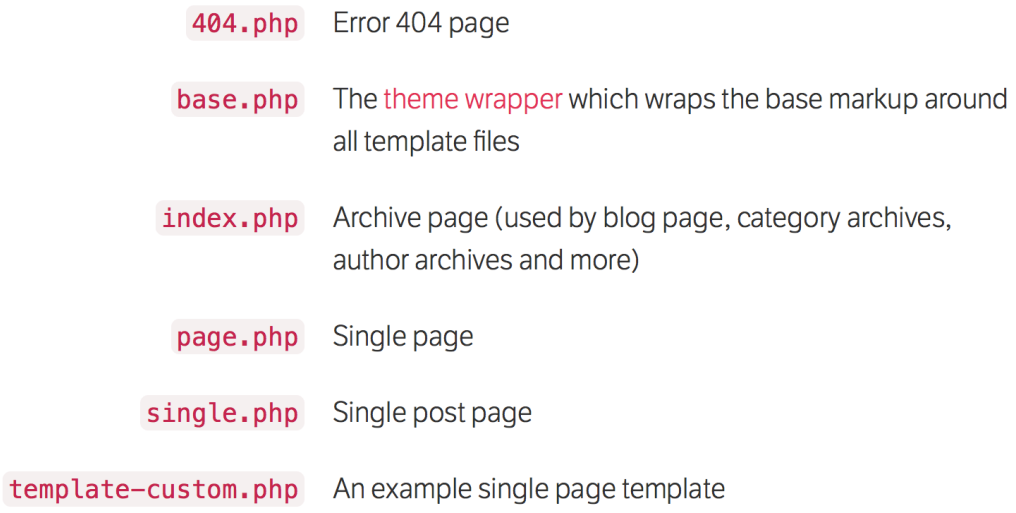
Jotta ruudukko saadaan rivittymään oikein, grid-sizer-luokalle kerrotaan, kuinka monta elementtiä mahtuu vierekkäin yhdelle riville. Suoja-alueiden koko määritellään gutter-sizer-luokan avulla. Arvot ja mediakyselyt liitetään tyylitiedostoon.

Kosketusnäytöllisiä laitteita varten jouduin piilottamaan videot ja korvaamaan ne kuvilla, koska hover-toiminto on suunniteltu vain hiiren osoittimelle.

7.2 Teema

Valitsin sivuston Wordpress-julkaisujärjestelmän pohjaksi Rootsin Sage-teeman. Sage on ilmainen aloitusteema, joka sopii erittäin hyvin räätälöidyn teeman rakentamiseen, koska siinä ei ole tyylejä tai sivupohjia valmiina. Ulkoasu ja sivupohjat täytyy luoda tyhjästä. Mallitiedostojen merkintäkieli perustuu HTML5 Boilerplate-standardiin, ja käytössä on myös osittain Bootstrapin elementtejä ja tyylejä. Tyylitiedostot kirjoitetaan Sass-kielellä, jolloin tiedostot pystytään jakamaan komponenttien mukaan. Myöhemmin ne tosin konvertoidaan yhteen tiedostoon Cascading style sheets -formaattiin. Tämä helpottaa muokkaamista ja työnkulkua. Lisäksi Sage sisältää modernit web-kehitys työkalut kuten gulpin, Bowerin ja BrowserSyncin, jolloin kehittäminen monelle laitteille on huomattavasti helpompaa ja nopeampaa. [38.]

Kuvassa 16 on teeman juurikansion sisältämiä tiedostoja, joita pystyy laajentamaan tavallisen Wordpress-hierarkian mukaisesti. [39.]



Kuva 16. Sage teeman juuritiedostot.

Näiden tiedostojen lisäksi teema sisältää templates-kansion, johon suurin osa räätälöinnistä päätyy [39]. Kuva 17 havainnollistaa templates-kansion sisällön.

`comments.php`	Markup for comments
`content-page.php`	Markup included from `page.php`
`content-single.php`	Markup included from `single.php`
`content.php`	Markup included from `index.php`
`entry-meta.php`	Post entry meta information included from `content-single.php`
`footer.php`	Footer markup included from `base.php`
`head.php`	`<head>` markup included from `base.php`
`header.php`	Header markup included from `base.php`
`page-header.php`	Page title markup included from most of the files in the theme root
`sidebar.php`	Sidebar markup included from `base.php`

Kuva 17. Sage teeman templates-kansion sisältö..

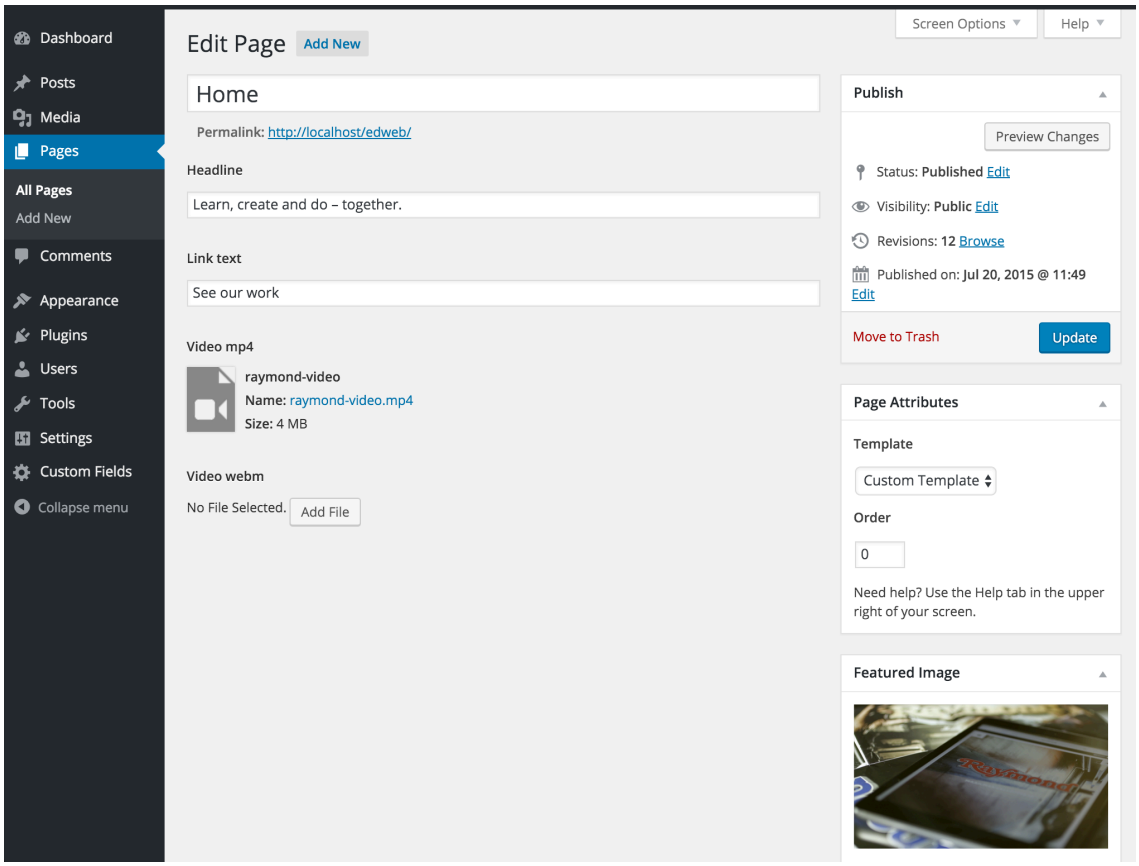
Tyylitiedostot ja JavaScript-tiedostot sijoitetaan asset-kansioon.

7.3 Teeman räätälöinti ja hallinta

Teemaa pystytään laajentamaan yleistä Wordpress-hierarkiaa seuraamalla, esimeriksi silloin, kun halutaan räätälöidä About-sivua. Pohjaksi kopioidaan page.php-tiedoston sisältö ja nimetään uusi tiedosto page-about.php. Tämän jälkeen About-sivu on muokattavissa sellaiseksi kuin halutaan. [40.]

Ensimmäisenä toteutin etusivun pohjatiedoston, joka käyttää tiedostoa template-custom.php. Sivupohja tulostaa otsakkeen ja silmukan, joka listaa 30 uusinta artikkelia packery-ruudukkoon. Otsakkeeseen voi ladata joko kuvan tai videon. Lisäsin otsakkeeseen videon lisäksi myös kentät Headline ja Link text käyttämällä Advanced Custom Field -liitännäistä. Body-kentän otin kokonaan pois. Kuva 18 konkretisoi, miten kentät näkyvät hallintapaneelissa. Otsakkeen korkeus, joka saadaan selainikkunan korkeu-

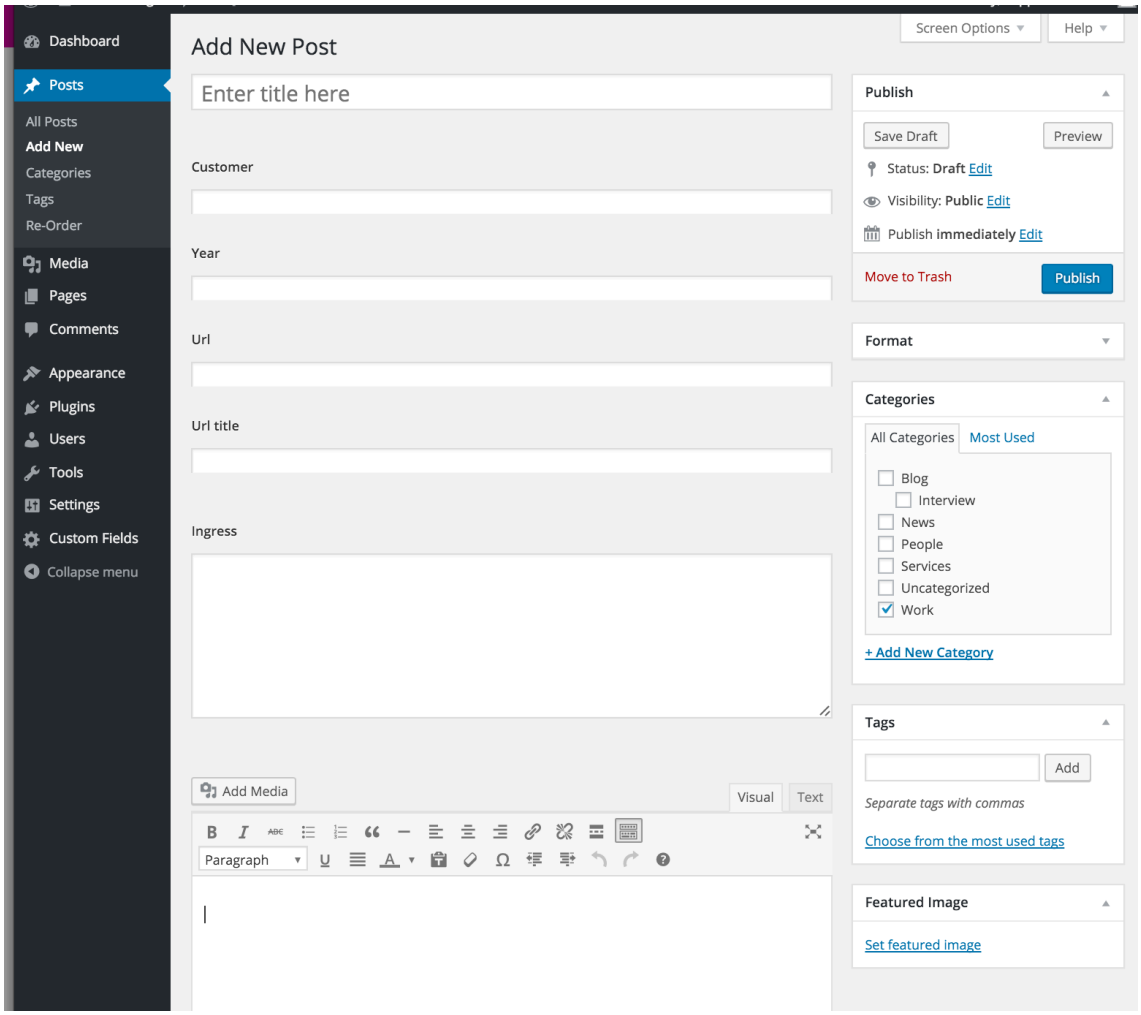
desta, asetetaan Jquery-funktiolla otsakkeen isäntäelementille. Korkeudessa täytyy kuitenkin huomioida suoja-alue eli gutter ja navigaatioelementin korkeus. Tästä saadaan seuraavanlainen yhtälö: selainikkunan korkeus - (suoja-alue + navigaation korkeus). Yhtälö lasketaan JavaScript-tiedossa. Sama toiminto lasketaan myös funktiossa, joka tarkastaa selainikkunan leveyden muutoksia. Jos otsakkeessa on video, se toistetaan automaattisesti. Etusivun lisäksi otsake asettuu ja toimii samalla tavalla myös artikkeli-sivuilla. Kosketusnäytöllisissä laitteissa otsakkeen video korvataan kuvalla, koska laitteet eivät tue auto play -ominaisuutta.



Kuva 18. Etusivun hallintanäkymä.

Artikkeleita hallinnoidaan julkaisujärjestelmän hallintapaneelissa post-välilehdellä, josta myös eri kategoriat luodaan. Artikkelit eroavat hieman toisistaan riippuen kategoriasta. Yhteistä kaikissa on, että ne sisältävät kentät ingressi, runko eli body, kuva ja video. Works- ja People-kategoriaan lisättiin myös muita tulostettavia kenttiä. Works-kategoriassa ylimääräiset kentät tulostetaan sivupalkkiin. Nämä lisäkentät ovat Customer, Year, Url ja Url title. People-kategoriassa ylimääräiset kentät tulostetaan otsakkeeseen. Kentät ovat Job title, Email, Phone, Linkedin profile ja Twitter profile. Artikke-

lisivut tulostetaan käyttöliittymään käyttämällä content-single.php-tiedostoa. Riippuen artikkelin kategoriasta sisällön jälkeen sivu näyttää aiheeseen liittyvää sisältöä. Esimerksi People-artikkelisivulla aiheeseen liittyvä sisältö on kaikki muut People-artikkelit. Aiheeseen liittyvillä sisällöillä on omat tiedostonsa, joissa on lähes samanlainen silmukka kuin etusivulla. Kuvassa 19 on kuvakaappaus Works-artikkelin hallintanäkymästä.



Kuva 19. Works-artikkelin hallintanäkymä..

Kategoriasivut listaavat yksittäisen kategorian artikkelit Packery-ruudukkoon. Kategorian kuvauksia hallinnoidaan Post-välilehdellä, josta valitaan Categories-osio. Sieltä käsin kategorioita voi luoda, muokata ja poistaa. Kategoria sivut käyttävät index.php-tiedostoa (kuva 20), jossa se tulostaa otsakkeen ja ruudukon. Kategoriasivuissa otsakkeessa on vain otsikko ja kuvaus.

```php
<?php get_template_part('templates/page', 'header-category'); ?>

<?php if (!have_posts()) : ?>
  <div class="alert alert-warning">
    <?php _e('Sorry, no results were found.', 'sage'); ?>
  </div>
  <?php get_search_form(); ?>
<?php endif; ?>

<section id="grid">
    <div class="grid-sizer"></div>
    <div class="gutter-sizer"></div>
    <div class="grid-item-ratio"></div>
    <?php while (have_posts()) : the_post(); ?>
      <?php get_template_part('templates/content', get_post_type() != 'post' ? get_post_type() : get_post_format()); ?>
    <?php endwhile; ?>
</section>

<?php the_posts_navigation(); ?>

```

Kuva 20. index.php-tiedosto.

7.4 Lisäosat

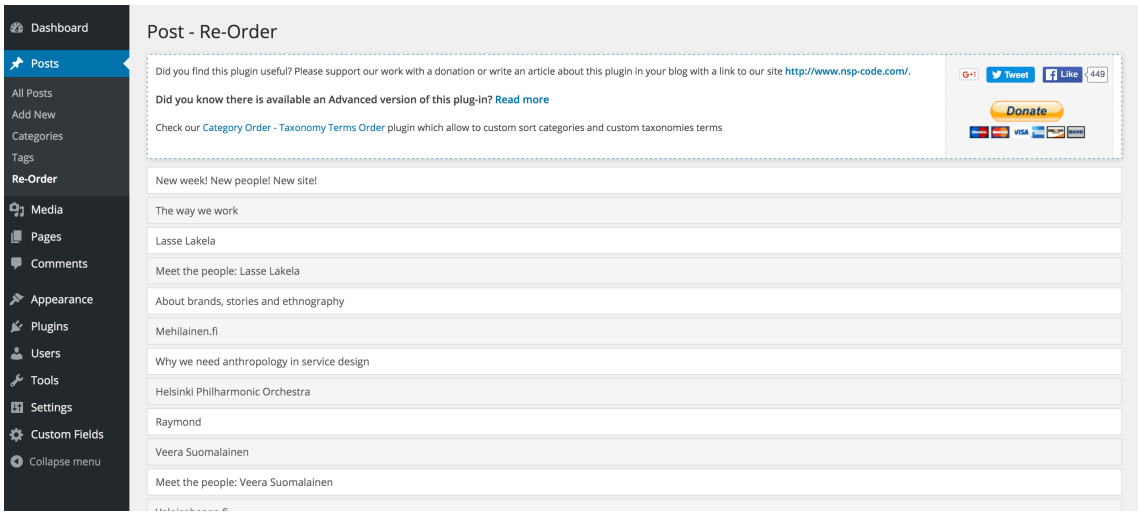
Opinnäytetyön sivustolla ovat käytössä lisäosat Advanced Custom Fields ja Post types order.

Advanced Custom Fieldin avulla pystytään luomaan omia kenttiä hallintanäkymään, jolloin sisällönhallinnasta saadaan käytettävämpää ja monipuolisempaa. Mallitiedostoissa kenttien arvot tulostetaan käyttämällä the_field()-funktiota. Kuva 21 osoittaa, kuinka funktiota voidaan käyttää. Sisällön ja sisällönrakenteen takia liitännäinen on välttämätön sivustolle. [41.]

```php
<?php

/**
 * Template Name: Home Page
 */

get_header();

?>

<div id="primary">
    <div id="content" role="main">

        <?php while ( have_posts() ) : the_post(); ?>

            <h1><?php the_field('custom_title'); ?></h1>

            <img src="<?php the_field('hero_image'); ?>" />

            <p><?php the_content(); ?></p>

        <?php endwhile; // end of the loop. ?>

    </div><!-- #content -->
</div><!-- #primary -->

<?php get_footer(); ?>
```

Kuva 21. Advanced Custom Field esimerkki.

Post type order on lisäosa, jossa sisältöä järjestellään raahaamalla se haluttuun järjestykseen Re-order-näkymässä (kuva 22). Julkaisujärjestelmä oletuksena järjestää sisällön aika tai aakkosjärjestykseen. Asensin lisäosan, koska etusivulla sisältöä täytyy tarvittaessa pystyä järjestämään räätälöidysti. Muilla sivuilla sisältö listataan artikkelin julkaisuajan mukaan uusimmasta alkaen. [42.]



Kuva 22. Re-order hallintanäkymä.

7.5 Hallittu lataus

Sivustolla on käytössä Lazy Load XT Jquery-liitännäinen, jolla pystytään säätelemään kuvien ja videoiden lataamista. Kirjaston avulla sivun lataamisesta saadaan nopeampaa, koska selain ei lataa kuvia tai videoita, ennen kuin ne ovat näkyvissä selainikkunassa. Vaikka sivulla olisi yhteensä 300 kuvaa, niistä ladataan vain ne jotka kulloinkin näkyvät selainikkunassa. Liitännäistä kutsuessa voidaan määritellä tarkemmin milloin selain aloittaa kuvien ja videoiden lataamisen. Asetin edgeY arvoksi 200, mikä tarkoittaa, että selain aloittaa kuvien tai videoiden lataamisen 200 pikseliä ennen, kuin selainikkunan alareuna on elementin kohdalla. Lisäksi muutin elementtien src-attribuutit muotoon “data-src”, jotta funktio toimii elementeissä. Lazy loading -liitännäinen oli välttämätön, koska sivusto sisältää paljon videoita ja isoja kuvia. Kuva 23 havainnollistaa, kuinka funktiota kutsutaan JavaScript-tiedostossa.

```
// LAZYLOAD
$.extend($.lazyLoadXT, {
  edgeY:  200,
  srcAttr: 'data-src'
});
```

Kuva 23. Lazy loadin kutsuminen JavaScript-tiedostossa.

7.6 Tulokset

Opinnäytetyön sivusto testattiin seuraavilla laitteilla iPhone 5, iPhone 6, iPad Air ja Macbook Pro, sekä virtuaalilaitteilla Nokia Lumia 930, Nokia Lumia 630, Samsung Galaxy S6 ja Windows 10. Tavoitteena oli selvittää, miten sivusto latautuu, toimiiko ruudukko ja onko mukautuvissa tyyleissä virheitä. Muutamia CSS-tyylivirheitä lukuun ottamatta sivustolla ei havaittu suurempia häiriöitä tai hidastelua. Laitetestauksen ohella suoritettiin myös Google PageSpeed Insights -testi. Tietokoneella yhteenvedoksi saatiin 62/100. Mobiililaitteella nopeudeksi saatiin 54/100 ja käyttökokemukseksi 100/100. Lisäksi selvitettiin testi What does my site cost? Se on palvelu, joka kertoo, kuinka raskas sivusto on ja kuinka kallista sitä on käyttää eri maissa mobiililiittymällä. Sivuston koko oli 10,68 megatavua.

Kuvien ja videoiden optimointia ja tiedostojen pakkaamista voisi siis vielä kehittää, jotta sivustosta saataisiin entistä nopeampi ja mobiiliystävällisempi. Kuviin voitaisiin esimerkiksi soveltaa PictureFill-toiminnallisuutta. Lataamisaikoja saataisiin nopeammiksi myös tehokkaammalla lazy loading -tekniikalla. Kuvien ja videoiden sijasta kokonaisia osioita voitaisiin ladata ja poistaa sitä mukaan, kuin käyttäjä selaa sivua.

Mukautuvuus toimii muuten todella hyvin ja sivusto näyttää suunniteltujen pohjien mukaiselta kaikissa testatuissa laitteissa. Kuvat skaalautuvat oikein, teksti on luettavaa ja ruudukossa kaikki on tasaista riippumatta laitteesta. Joitakin kuvasuhdevirheitä on havaittavissa videoissa kun selainikkunan kokoa muutetaan raahaamalla. Jostain syystä selain ei ehdi tarkistaa, onko videon kuvasuhde isompi kuin sen isäntäelementti, johon videon pitäisi mukautua. Jos video on matalampi kuin isäntä elementti, se venytetään yhtä leveäksi. Jos video on leveämpi sen korkeutta taas manipuloidaan. Virheen huomaa kun videon reunalla näkyvät mustat palkit.

Toinen jatkokehitystä ja selvitystä vaativa asia on videoiden toistaminen kosketusnäytöllisissä laitteissa. Tällä hetkellä videot piilotetaan niissä laitteissa, joissa on ominaisuutena kosketusnäyttö, ja tilalla näytetään kuva. Hover-toimnnon sijasta video voitaisiin toistaa silloin, kun selainikkuna on videon kohdalla. Samanlainen toiminnallisuus on käytössä esimerkiksi Facebookin mobilisovelluksessa. Suurin ongelma on kuitenki se, että videoita ei saada elementtien taustalle toistumaan. Esimerksi iPhone ja iPad avaavat videot niiden omassa videosovelluksessa eivätkä toista niitä suoraan se-

laimessa. Vaihtoehtoinen ratkaisu voisi olla korvata videot png-sekvensseillä eli sarjoina kuvia, joita näytetään peräkkäin.

Sivuston saama palaute on ollut positiivista, ja varsinkin visuaalinen ilme on saanut paljon kiitosta. Videoiden puuttuminen kosketusnäytöissä on harmittanut muutamaa kävijää.

8 Yhteenveto

Insinöörityössä suunniteltiin ja toteutettiin verkkosivusto, jossa on käytössä uusimpia verkkokehityksen ja verkkosuunnittelun ratkaisuja. Raportissa selvitettiin kuinka sivustolle saatiin luotua modulaarinen ruudukko selainpuolen sovelluskehyksiä ja JavaScript-kirjastoja hyväksikäyttämällä. Sivuston käyttöliittymän suunnitelma pohjautuu tarkkaan ruudukkoon, joka on modulaarinen ja mukautuva. Käyttöliittymä rakentuu komponenteista, joiden esitystapa ei riipu sivusta tai sivupohjasta. Niiden ulkonäkö on siis ennalta määritelty ja pysyy samana sivusta riippumatta. Esitystavat ja toiminnot standardisoitiin, jotta jokainen komponentti näkyi sivustolla oikein. Käyttöliittymä mukautuu mobiili ensin -periaatteella ominaisuuksien mukaan ja tarjoaa parhaan mahdollisen käyttökokemuksen laitteesta riippumatta.

Sivusto rakennettiin Wordpress-julkaisujärjestelmän päälle, jolla hallittaan ja päivitetään käyttöliittymässä näkyvää sisältöä. Julkaisujärjestelmän teemaksi valittiin aloitusteema, joka räätälöitiin sisältöjen ja käyttötapojen mukaan. Sisällönhallinnan helpottamiseksi julkaisujärjestelmään asennettiin myös tarvittavat lisäosat.

Exove Design yrityksen verkkosivujen toteutus ja suunnittelu oli mielenkiintoinen työ, ja siinä oli sopivasti haastetta. Tavoitteena oli toteuttaa ajankohtainen ja räätälöity sivusto, jossa sovelletaan modulaarista ajattelutapaa sen ruudukossa ja elementeissä. Ruudukko toteutettiin käyttämällä Bootstrap sovelluskehyksen mukautuvuutta, ja elementit kohdistettiin paikalleen Packery JavaScript-kirjastolla. Tuloksena saatiin eheä ja käytettävä sivusto, joka testattiin toimivaksi mobiilissa, tabetissa ja tietokoneella. Käyttökokemuksessa sivusto sai täydet pisteet Googlen PageSpeed insight -testissä. Sivuston nopeudessa ja kuvien sekä videoiden optimoimisessa tosin olisi vielä paljon parannettavaa. Tiedostoja voidaan pakata pienemmäksi ja siirtää välimuistiin. Tulevaisuudessa tutkimusta ja sivuston kehitystä voidaan jatkaa varsinkin mobiililaitteissa, Kuinka esi-

merkiksi saadaan videot toistumaan niissä ilman, että ne toistuvat suoraan laitteiden omissa sovelluksissa. Tällä hetkellä videoiden toistuminen HTML-elementtien taustalla suurimmalla osalla mobiililaitteista ei ole vielä mahdollista. Kokonaisuudessaan insinöörityö vastasi sille asetettuja tavoitteita. Työ oli merkityksellinen ammatillisen kehitykseni kannalta ja opin hyödyntämään uusia työkaluja verkkosuunnittelussa.

Lähteet

1 Web design. 2016. Verkkodokumentti. Wikipedia <https://en.wikipedia.org/wiki/Web_design>. Päivitetty 28.3.2016. Luettu 30.3.2016.

2 Cao, Jerry 2015. 6 Web Design Trends You Must Know for 2015 & 2016. Verkkodokumentti <http://www.awwwards.com/6-web-design-trends-you-must-know-for-2015-2016.html> Päivitetty 16.9.2015. Luettu 30.3.2016

3 Makino, Takaki & Phan, Doantam. 2015. Rolling out the mobile-friendly update. Verkkodokumentti. Rolling out the mobile-friendly update. <https://webmasters.googleblog.com/2015/04/rolling-out-mobile-friendly-update.html>. Päivitetty 21.4.2015. Luettu 30.3.2016.

4 Wroblewski, Luke. 2012. Which One: Responsive Design, Device Experiences, or RESS? Verkkodokumentti. <http://www.lukew.com/ff/entry.asp?1509> Päivitetty 29.2.2012. Luettu 30.3.2016.

5 Louie, A. 2014. Why and How to Avoid Hamburger Menus. Verkkodokumentti. <https://lmjabreu.com/post/why-and-how-to-avoid-hamburger-menus/> Päivitetty 14.5.2014. Luettu 30.3.2016.

6 Lessin, Jessica E. 2013. What's a Hamburger button? A Guide to App Features. Verkkodokumentti. <http://blogs.wsj.com/digits/2013/03/18/whats-a-hamburger-button-a-guide-to-app/> Päivitetty 18.3.2013. Luettu 30.3.2016.

7 Fadeyev, Dmitry. 2013. Authentic Design. Smashing magazine. Verkkodokumentti. <https://www.smashingmagazine.com/2013/07/authentic-design/> Päivitetty 16.7.2013. Luettu 30.3.2016.

8 Clum, Luke. 2016. The beginner's guide to flat design. Verkkodokumentti. <http://www.creativebloq.com/graphic-design/what-flat-design-3132112> Päivitetty 10.3.2016. Luettu 30.3.2016.

9 Knight, Kayla. 2011. Responsive web design: what it is and how to use it. Verkkodokumentti. Smashing magazine. <http://coding.smashingmagazine.com/2011/01/12/guidelines-for-responsive-web-design/> Päivitetty 12.1.2011. Luettu 30.3.2016.

10 Marcotte, Ethan. 2011. Responsive web design. New York: A Book Apart.

11 Rutter, Richard. Choose a comfortable measure. Verkkodokumentti. <http://webtypography.net/2.1.2> Luettu 30.3.2016.

12 Marquis, Mat. 2012. Responsive images: How they almost worked and what we need. Verkkodokumentti. A list apart. <http://www.alistapart.com/articles/responsive-images-how-they-almost-worked-and-what-we-need/> Päivitetty 31.12.2012. Luettu 30.4.2016.

13 Responsive images: Experimenting with context-aware image sizing. 2010. Verkkodokumentti. Filament group. <http://filamentgroup.com/lab/responsive_images_experimenting_with_context_aware_image_sizing/> Päivitetty 14.12.2010. Luettu 30.3.2016.

14 Picurefill 2.3.1. Verkkodokumentti. Responsive design. <https://responsivedesign.is/resources/images/picture-fill> Luettu 30.4.2016.

15 Wroblewski, Luke. 2011. Mobile first. New York, New York: A Book Apart.

16 Shillcock, Rachel. 2013. All About Grid Systems. Verkkodokumentti. Envato. <http://webdesign.tutsplus.com/articles/all-about-grid-systems--webdesign-14471> Päivitetty 22.8.2013. Luettu 30.3.2016.

17 Hampton-Smith, Sam. 2014. The designer's guide to grid theory. Verkkodokumentti. Creative Blog. <http://www.creativebloq.com/web-design/grid-theory-41411345> Päivitetty 15.4.2014. Luettu 30.3.2016.

18 Responsive Web Design - Grid-View. Verkkodokumentti. W3Schools. <http://www.w3schools.com/css/css_rwd_grid.asp> Luettu 30.3.2016.

19 Oladunni, Olawale. 2013. Mojo Motors' Responsive Redesign With Fireworks: UX And Interaction Design. Verkkodokumentti. Smashing Magazine. < http://www.w3schools.com/css/css_rwd_grid.asp> Päivitetty 26.8.2013. Luettu 30.3.2016.

20 Boulton, Mark. 2005. Five simple steps to designing grid systems. Verkkodokumentti. <http://markboulton.co.uk/journal/five-simple-steps-to-designing-grid-systems-part-4> Päivitetty 30.8.2005. Luettu 30.3.2016.

21 Croft, Jeff. 2007. Framework for Designers. Verkkodoumentti. A list Apart. <http://alistapart.com/article/frameworksfordesigners> Päivitetty 12.6.2007. Luettu 30.3.2016.

22 What are Frameworks? 22 Best Responsive CSS Frameworks for Web Design. Verkkodokumentti. Awwwards. <http://www.awwwards.com/what-are-frameworks-22-best-responsive-css-frameworks-for-web-design.html> Luettu 30.3.2016.

23 Gerchev, Ivaylo. 2014. The 5 Most Popular Frontend Frameworks of 2014 Compared. Verkkodokumentti. Sitepoint. <http://www.sitepoint.com/5-most-popular-frontend-frameworks-compared/> Päivitetty 11.12.2014. Luettu 30.4.2016.

24 Bootstrap. Verkkodokumentti. Bootstrap. <http://getbootstrap.com/> Luettu 30.4.2016.

25 CSS. Verkkodokumentti. Bootstrap. <http://getbootstrap.com/css> Luettu 30.4.2016.

26 Covier, Chris. 2008. What Are The Benefits of Using a CSS Framework? CSS Tricks. <https://css-tricks.com/what-are-the-benefits-of-using-a-css-framework/> Päivitetty 22.10.2008. Luettu 30.3.2016.

27 Andrew, Paul. 2011. Discussing the Pros and Cons of Using a CSS Framework. Verkkodokumentti. Specky Boy. <https://speckyboy.com/2011/03/14/discussing-the-pros-and-cons-of-using-a-css-framework/> Päivitetty 14.3.2011. Luettu 30.3.2016.

28 Wordpress. Verkkodokumentti. Wordpress <https://wordpress.org/> Luettu 30.3.2016.

29 CMS Usage Statistics. Verkkodokumentti. Built with. <http://trends.builtwith.com/cms> Luettu 30.3.2016.

30 Theme Development. Verkkodokumentti. Wordpress. <https://codex.wordpress.org/Theme_Development> Luettu 30.3.2016.

31 Child Themes. Verkkodokumentti. Wordpress.<https://codex.wordpress.org/Child_Themes> Luettu 30.3.2016.

32 Plugins. Verkkodokumentti. Wordpress <https://wordpress.org/plugins/browse/popular/> Luettu 30.3.2016.

33 Writing a Plugin. Verkkodokumentti. Wordpress. <https://codex.wordpress.org/Writing_a_Plugin> Luettu 30.3.2016.

34 Pros and Cons of Wordpress. Verkkodokumentti. Web builders guide. <http://www.webbuildersguide.com/website-builder-articles/pros-and-cons-of-wordpress/> Luettu 30.3.2016.

35 Isotope. Verkkodokumentti. Metafizzy. <http://isotope.metafizzy.co/layout-modes/masonry.html> Luettu 30.3.2016.

36 Masonry. Verkkodokumentti. David DeSandro. <http://masonry.desandro.com/> Luettu 30.3.2016.

37 Packery. Verkkodokumentti. Metafizzy. <http://packery.metafizzy.co/> Luettu 30.3.2016.

38 Roots Sage. Verkkodokumentti. Roots. <https://roots.io/sage/> Luettu 30.3.2016.

39 Theme Templates. Verkkodokumentti. Roots <https://roots.io/sage/docs/theme-templates/> Luettu 30.3.2016.

40 Theme Wrapper. Verkkodokumentti. Roots <https://roots.io/sage/docs/theme-wrapper/> Luettu 30.3.2016.

41 Displaying values in your theme. Advanced Custom Field. <https://www.advancedcustomfields.com/resources/displaying-custom-field-values-in-your-theme/> Luettu 30.3.2016.

42 Post types order. Verkkodokumentti. Wordpress. <https://wordpress.org/plugins/post-types-order/> Luettu 30.3.2016