

KARELIA-AMMATTIKORKEAKOULU  
Viestinnän koulutusohjelma

Tommi Valtonen

KARJALAINEN.FI-VERKKOPALVELUN FRAMEWORKIN  
UUDISTAMINEN

Opinnäytetyö  
Toukokuu 2017



**OPINNÄYTETYÖ**  
**Toukokuu 2017**  
**Viestinnän koulutusohjelma**

Länsikatu 15  
80110 JOENSUU  
(013) 260 600

Tekijä(t)  
Tommi Valtonen

Nimeke  
Karjalainen.fi-verkkopalvelun frameworkin uudistaminen.

Toimeksiantaja  
Pohjois-Karjalan Kirjapaino Oy

**Tiivistelmä**

Opinnäytteen aiheena on tehdä Sanomalehti Karjalaisen verkkopalvelun frameworkin uudistaminen. Frameworkilla tarkoitetaan ohjelmaa, jolla luodaan verkkopalvelun rakenne. Tavoitteena oli tehdä kopio Karjalaisen nykyisestä verkkopalvelusta uuteen frameworkiin, jonka avulla pyritään nopeuttamaan verkkopalvelun latausaikoja. Frameworkilla haluttiin myös helpottaa verkkopalvelun muokkaamista tulevaisuudessa.

Teoriaosuudessa kerron erilaisia verkkosuunnittelun keinoja, joita käytetään projektin aikana. Tietopohja perustuu alan kirjallisuuteen ja erilaisiin verkkosuunnittelun keinoihin, joita sovelletaan projektin aikana. Teoriaosuudessa kerrotaan myös frameworkin eri ominaisuuksista tarkemmin. Verkkopalvelun uutena frameworkina tullaan käyttämään Joomshaperin Helix3 -frameworkia. Frameworkia lähdettiin ensin työstämään Karjalaisen testiserverillä, jolle rakennetaan verkkopalvelusta kopio uuteen frameworkiin. Frameworkin valmistuttua julkaistiin uusi framework ja korjattiin mahdolliset ongelmat. Projektin aikana pyydettiin palautetta tehdystä työstä ja myös kehitysideoita projektin johdolta. Projektia avusti myös Karjalaisen yhteistyökumppani Fiare Oy.

Opinnäytetyön aikana onnistuttiin tuottamaan kopio Karjalaisen sivustosta uuteen frameworkiin ja nopeuttamaan sivuston latausaikaa kahdella sekunnilla. Opinnäytetyön loppupäätelmänä on kuitenkin, että palvelun latausaikaa voisi vielä nopeuttaa muilla keinoilla.

Kieli  
Suomi

Sivuja 28  
Liitteet 3  
Liitesivumäärä 8

Asiasanat  
CSS, responsiivinen verkkosuunnittelu, framework



**THESIS**  
**May 2017**  
**Degree Programme in communication**  
Tikkarinne 9  
FI 80200 JOENSUU  
FINLAND  
(013) 260 600

Author(s)

Tommi Valtonen

Title

Framework Update of Karjalainen.fi  
Commissioned by  
Pohjois-Karjalan Kirjapaino Oy

Abstract

The purpose of the thesis was to update the framework of karjalainen.fi website. Karjalainen.fi belongs to the regional paper Karjalainen and is the homepage of the newspaper. The goal was to produce a copy of current karjalainen.fi to the new framework. In doing so we are hoping to reduce the load times of karjalainen.fi. Also the new framework will make it easier to modify the web service in the future.

Theory part of the thesis describes different styles of web development, which will be used during the project. The theoretical background of the thesis is based on the literature of web development and different styles of web development, which are adapted during the project. The theory also explains the framework and its different functions in more detail. Joomla's Helix3 framework was used to create the new framework for the website. First of all we are going to start working on a test server, where we will produce a copy of the current karjalainen.fi. After the copy is ready, we will publish the new framework and fix any problems that might occur. Project management provided feedback during the project. Fiare Oy the business partner of Karjalainen was also part of the project.

During the thesis project we were able to produce a copy of karjalainen.fi to the new framework. We also managed to reduce the load times by two seconds. The result of the thesis is that we could reduce the load times more using other methods.

Language  
Finnish

Pages 28  
Appendices 3  
Pages of Appendices 8

Keywords

CSS, responsive web design, framework

## Sisältö

1	Johdanto .....	6
2	Opinnäytetyön tietoperusta .....	7
2.1	Joomla .....	7
2.2	Template .....	9
2.3	Framework .....	9
2.4	CSS .....	11
2.5	Moduulit .....	12
3	Opinnäytetyön lähtökohta ja menetelmälliset valinnat .....	13
3.1	Toimintaympäristö ja lähtötilanne .....	13
3.2	Toiminnan eteneminen .....	15
3.3	Navigaatio ja layout .....	16
3.4	Pääuutinen ja kärkiuutiset .....	18
3.5	Class-arvot ja kuvien optimointi .....	20
3.6	Menun käyttäminen tabletilla .....	21
3.7	Custom CSS-moduulit ja adblock .....	22
3.8	Julkaisu .....	23
3.9	Yhteenveto prosessista .....	24
4	Tulokset .....	25
5	Pohdinta .....	26
	Lähteet .....	29

### Liitteet

Liite 1	Alkuperäinen idea templatesta
Liite 2	Wireframe kuvat etusivu, kategoria ja artikkeli
Liite 3	Käsin piirretty malli adblock –moduulin toiminnasta

## Sanasto

CSS	CSS eli Cascading style sheets on verkkosivustojen tyyli-tiedosto, mikä määrittää sivuston eri elementtien tyylit. CSS:llä voidaan määrittää esimerkiksi fonttien koot, sekä erilaiset värimääritelmät. Myös elementtien kokoa ja asette-lua on mahdollista hallita CSS:llä. (Smith & Negrino 2007, 263.)
ELEMENTTI	Elementti on tässä opinnäytteessä yleisnimitys verkkopal-velussa oleville asioille. Esimerkiksi moduulissa oleva kuva on yksittäinen elementti. (W3 2014.)
FOOTER	Footerilla tarkoitetaan verkkosivuston alaosaa. Footer si-jaitsee muun materiaalin alapuolella. Footerissa säilytetään myös sivustoon liittyvää tietoa. (W3 2011b.)
FRAMEWORK	Frameworkilla tarkoitetaan ohjelmaa, jonka avulla on mah-dollista luoda helposti muokattava sivupohja eli template (Joomshaper 2016a).
HEADER	Headerilla tarkoitetaan verkkosivuston yläosaa, joka sijait-see muun materiaalin yläpuolella. Headerissa on yleensä sivuston logo ja navigaatio. (W3 2011a.)
JOOMLA	Joomla on avoimeen lähdekoodiin perustuva sisällönhallin-tajärjestelmä (Joomla 2017). Joomla'n kautta hallitaan verkkopalvelun sisältöä, sekä sivuston palvelun ulkoasua.
LAYOUT	Layout tarkoittaa sivuston pohjapiirrustusta, jossa määrite-tään elementtien sijainti verkkosivulla (Joomshaper 2016b).
MODUULI	Moduulien avulla tuodaan Joomla'n sisältöä esille sivustol-la. Moduulit asetetaan sivupohjassa määritettyihin alueisiin, joita kutsutaan moduulipaikoiksi. Sivupohja määrittää mis-säpäin sivua moduuleita pystyy näyttämään. (Repo 2012, 27.)
TEMPLATE	Templatella tarkoitetaan sivustolla käytettävää sivupohjaa. Sivupohjassa määritellään mihin paikkoihin sivustolla on mahdollista näyttää moduuleita. (Docs.Joomla 2014.)

## 1 Johdanto

Digitaaliset palvelut tuovat teknisiä haasteita julkaisijoilleen. Sanomalehti Karjalaisen omilla verkkosivuilla on ollut jo pidemmän aikaa ongelmana hidas latausaika. Sivujen latausajat ovat enimmillään olleet lähempänä kymmentä sekuntia, kun latausajan ei pitäisi ylittää viittä sekuntia. Ongelmaa on yritetty ratkaista Karjalaisen tietohallinnossa jo useamman kuukauden ajan. Ongelman ratkaisemiseksi päätettiin vaihtaa verkkosivun sivupohja kevyempään sivupohjaan. Uutena sivupohjana tullaan käyttämään Joomshaperin Helix3-frameworkia, jonka avulla voimme luoda oman sivupohjan. Helix3 on paljon kevyempi kuin Karjalaisen nykyinen sivupohja ja sitä on myös paljon helpompi muokata tarpeen vaatiessa.

Opinnäytteeni tarkoituksena on päivittää sanomalehti Karjalaisen verkkopalvelun framework. Verkkopalvelussa tullaan käyttämään jatkossa Joomshaperin Helix3-frameworkia. Frameworkilla tarkoitetaan ohjelmaa, jonka avulla on mahdollista luoda helposti muokattava sivupohja eli template (Joomshaper 2016a). Sivupohjassa määritellään, mihin paikkoihin sivustolla on mahdollista näyttää moduuleita (Docs.Joomla 2014). Frameworkin päivittämisellä pyritään siihen, että Karjalaisen verkkopalvelun latausaikoja saataisiin lyhennettyä, koska Helix3 on kevyempi ladattava kuin sivuston nykyinen sivupohja. Karjalainen on seitsemänä päivänä ilmestyvä sanomalehti, jonka kustantajana toimii Sanomalehti Karjalainen Oy (Mediamyynti.karjalainen 2017a). Karjalainen keskittyy pohjoiskarjalaisiin uutisiin ja on kahdennakseksi suurin maakuntalehti verkossa (Mediamyynti.karjalainen 2017b). Karjalaisen verkkopalvelu löytyy osoitteesta [www.karjalainen.fi](http://www.karjalainen.fi). Helix3-framework on jo käytössä Karjalaisen tytärlhdillä, joihin kuuluu Karjalan Heili, Outokummun Seutu, Lieksan Lehti, Pielisjokiseutu, Parikkala-Rautjärven Sanomat, Pogostan Sanomat ja Ylä-Karjala.

Projektin aikana tehdään kopio nykyisestä Karjalaisen sivustosta ja vaihdamme sivupohjan uuteen Helix3-frameworkiin. Vaihtamisen jälkeen rakennamme frameworkilla nykyistä Karjalaisen sivustoa vastaavan sivupohjan, jonka jälkeen kytkemme nykyiset moduulit päälle. Tämän jälkeen aloitetaan tyylitiedoston

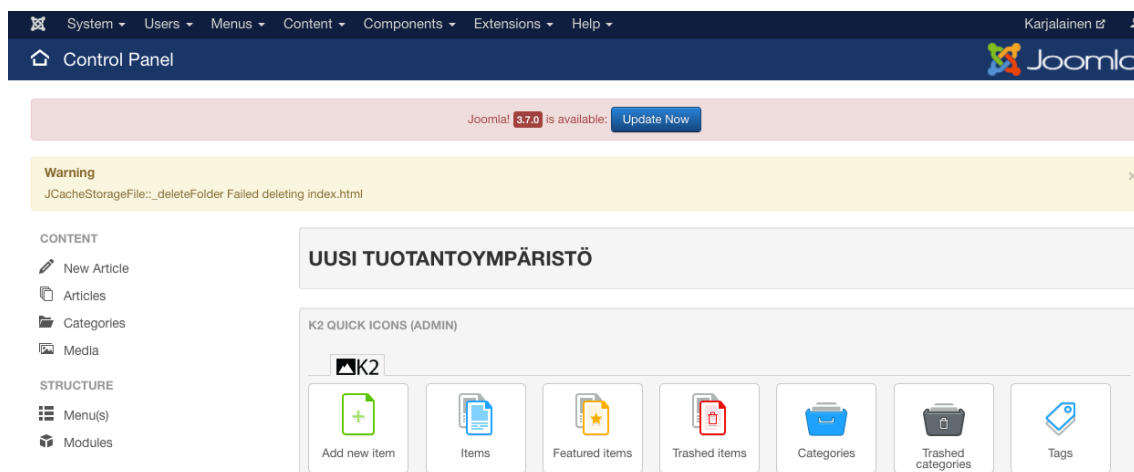
muokkaus, jotta sivusto saadaan muistuttamaan nykyistä Karjalaisen sivustoa. Opinnäytteen tarkoituksena on saada Karjalaisen verkkopalvelun latausaikoja vähennettyä, sekä luoda kopio nykyisestä sivustosta uudelle frameworkille. Uusi framework mahdollistaa myös sivuston ulkoasun helpomman muokkaamisen jatkossa Helix3:n layout ominaisuuden vuoksi. Opinnäytteen aikana tutkitaan millä keinoilla verkkopalvelun latausaikoja voidaan lyhentää, sekä millä keinoilla uusi framework helpottaa palvelun käyttöä ja kehittämistä.

## **2 Opinnäytetyön tietoperusta**

### **2.1 Joomla**

Suosituimpia julkaisujärjestelmiä ovat tällä hetkellä Joomla, Drupal ja WordPress. Drupal soveltuu monimutkaisten sivustojen tekoon ja on tehokkain näistä kolmesta sisällönhallintajärjestelmästä, mutta tarvitsee käyttäjä tarvitsee paljon teknistä osaamista, jotta hän voi hyödyntää Drupalin kaikkia ominaisuuksia. Joomla ja WordPress eivät vaadi käyttäjältään niin laajaa tietämystä verkkosivustojen teosta kuin Drupal. Joomla on avoimeen lähdekoodiin perustuva sisällönhallintajärjestelmä (Joomla 2017). Joomla ja WordPressin etuna Drupaliin on helppokäyttöisyys. Helppokäyttöisin näistä kolmesta on varmasti WordPress, mutta Joomla etu WordPressiin nähden on se, että sillä voi luoda monimutkaisempia verkkosivuja hyvin vähäisellä opettelulla. Joomlailla on myös laaja verkosto, josta löytyy apua ongelmatilanteissa. (Saajola 2013, 7-8.)

Joomla antaa käyttäjälleen mahdollisuudet muokata sivunsa sisältöä vaivattomasti ja se tekee myös verkkopalvelun ylläpitämisestä helppoa (Kuva 1). Ohjelmistojen päivitys käy helposti ja sisällön tuottamiseen ei tarvita erityisosaamista. (Kuusk 2016, 13.)



Kuva 1. Joomla:n käyttöjärjestelmä.

Joomla sopii mielestäni hyvin kyseisen opinnäytteen sisällönhallintajärjestelmäksi, koska sillä voi toteuttaa monipuolisia sivuja. Näistä kolmesta sisällönhallintajärjestelmästä minulla itselläni on eniten kokemusta Joomla:sta. Drupalia en ole päässyt testaamaan, mutta olen tehnyt jonkin verran WordPress-sivuja. WordPress soveltuu mielestäni muutaman sivun kokoiseen verkkopalveluun mainiosti, mutta monipuolisemman palvelun toteuttaminen olisi haastavaa. Itselläni vähän on kokemusta WordPressin ilmaisversiosta, joka on mielestäni hyvä työkalu perinteisen verkkopalvelun perustamiseen. Monimutkaisemman verkkopalvelun toteuttamista varten tarvittaisiin kuitenkin maksullisen version tarjoamia työkaluja. Joomla:lla on kuitenkin paljon ilmaisia työkaluja ja lisäosia, joiden avulla monipuolisen verkkopalvelun toteuttaminen on helpompaa.

Jotkut tutkimukset kuitenkin painottavat, että Joomla:n käyttämiseen tarvitaan tuskin ollenkaan kokemusta verkkopalvelun tekemisestä. Omasta mielestäni tämä ei pidä täysin paikkaansa. Esimerkiksi ulkoasun ja lisäosien ulkoasun muokkaamiseen tarvitaan aika paljon tietoa verkkopalvelun suunnittelusta. Omasta mielestäni jo pelkkä CSS-koodin osittainen hallitseminen on pakollista, kun verkkosivuja lähdetään suunnittelemaan. Myös ohjelmistojen päivittäminen vaatii aika paljon ammatillista osaamista asiasta, jotta palvelu on tietoturvallinen. Ohjelmien päivitys tapahtuu kyllä muutamaa nappia painamalla tai lataamalla muutama tiedosto toisista palveluista, mutta päivittämisen jälkeen on mahdollista, että jotkin palvelun lisäosat eivät toimi nykyisen version kanssa. Joomla:n käyttöjärjestelmään tottuu kyllä nopeasti ja sisällöntuottaminenkin onnistuu helposti, jos on rakentanut yhden toimivan ratkaisun, jota voi monistaa.



## 2.2 Template

Templatella tarkoitetaan sivustolla käytettävää sivupohjaa, johon sivusto rakennetaan. Sivupohjassa määritellään, mihin paikkoihin sivustolla on mahdollista näyttää moduuleita (Docs.Joomla 2014). Templatea voi muokata eri tavoilla, kuten esimerkiksi CSS:llä. CSS:llä voi muokata sivupohjan tyyliä ja moduulien asettelua, kun taas PHP:llä voi lisätä moduulipaikkoja tai muokata niitä haluamallaan tavalla. Joomla:ssa itsessään on jo muutama sivupohja valmiina asennettaessa, mutta sivupohjia voi ladata kuitenkin verkosta lisää palveluista kuten [extensions.joomla.org](http://extensions.joomla.org). (Oinonen 2013, 23.) Valmiilla sivupohjilla on mahdollista toteuttaa helposti toimiva ja näyttävä verkkopalvelu. Avoin lähdekoodi tarjoaa myös avaimet palvelun laajempaan ulkoasulliseen muokkaamiseen. (Repo 2012, 32-33.)

Olen samaa mieltä aiempien havaintojen kanssa siitä, että valmiilla sivupohjalla on mahdollista toteuttaa toimiva ja näyttävä verkkopalvelu. Valmiit sivupohjatkin kuitenkin tarvitsevat jonkin verran ammatillista osaamista, jotta sivupohjasta saa todella näyttävän. Myös sivustolla näytettävät moduulit tarvitsevat usein ulkoasun muokkausta, koska ne eivät välttämättä asetu oikein kaikille sivupohjille. Nykyään on myös muistettava responsiivisuus, koska sivupohjaa voidaan maistaa responsiiviseksi, mutta yleensä sivupohjan CSS-tiedostoon pitää tehdä tarvittavia muutoksia, jotta sivusto näyttää hyvältä mobiilissa ja tietokoneella.

Omasta mielestäni valmista sivupohjaa on hyvä käyttää pohjana verkkopalvelulle. Palvelusta saadaan kuitenkin kaikki mahdollinen hyöty irti, jos palvelu toimii hyvin kaikilla laitteilla. Valmiit sivupohjat eivät myöskään yleensä vastaa täydellisesti asiakkaan näkemystä verkkopalvelustaan, joten tästä syystä on suositeltavaa, että verkkopalvelun tekijällä on tietoa ulkoasun muokkaamisesta.

## 2.3 Framework

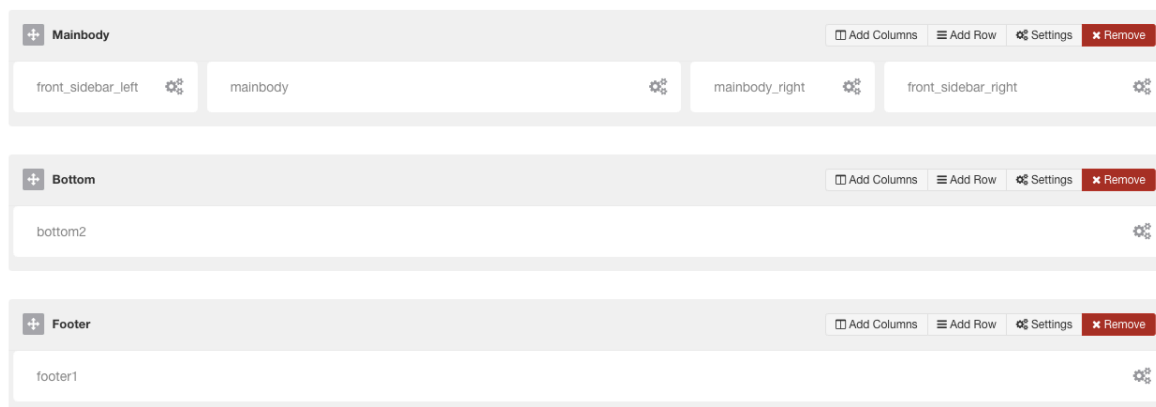
Frameworkilla tarkoitetaan ohjelmaa, jonka avulla on mahdollista luoda helposti muokattava sivupohja eli template (Joomshaper 2016a). Tavallisesti sivupohjis-

sa on jo valmiiksi määritelty alueet minne voidaan kytkeä esimerkiksi moduuleita. Helix3-frameworkissa voidaan kuitenkin rakentaa sivuston rakenne layout työkalun avulla. Layout tarkoittaa sivuston pohjapiirustusta, jossa määritetään mitä missäkin näytetään (Joomshaper 2016b). Helix3-frameworkin tekemiseen on käytetty Bootstrap 3-frameworkia, joka tekee sivupohjasta responsiivisen (Joomshaper 2016a).

Bootstrapin sarakkeet helpottavat varsinkin graafikon työtä. Sarakkeet tekevät ulkoasun suunnittelusta helpompaa, kun graafikko voi sarakkeiden avulla suunnitella eri elementtien leveyksiä. (Blomberg 2015, 36.) Bootstrap käyttää 12-sarakkeista jakoa (Blomberg 2015, 33). Tällä tarkoitetaan sitä, että sivusto jaetaan 12 sarakkeeseen, joiden kokoa on mahdollista muuttaa. Sarakkeet voi jakaa esimerkiksi 2, 8, 2 jolloin keskelle tulee yksi iso sarake ja sivuille pienemmät sarakkeet.

Responsiivinen framework mahdollistaa sen, että verkkosivua voi käyttää niin mobiililaitteella, kuin tietokoneellakin. Responsiivisuuden tekeminen ja suunnittelu vie enemmän aikaa, mutta lisää palvelun tehokkuutta. (Blomberg 2015, 36.) Responsiivisuus on nykypäivisin hyvin tärkeä osa verkkopalveluiden suunnittelua. Laitteet kehittyvät jatkuvasti, joten sivustoa suunnitellessa on hyvä keskittyä mahdollisimman paljon siihen miten palvelu toimii erikokoisilla laitteilla. Mielestäni Bootstrap framework helpottaa verkkosivujen suunnittelua huomattavasti, koska sarakkeiden avulla on helppo suunnitella eri elementtien kokoja verkkopalvelussa.

Helix3 sisältää hyvin yksinkertaisen ja selkeän layout-työkalun, jonka avulla Bootstrap sarakkeiden suunnittelu käy helposti (Kuva 2). Vaikka Bootstrap helpottaakin suunnittelua se myös lisää työmäärää huomattavasti. Responsiivisuuden suunnittelu onnistuu Bootstrapin avulla helposti, mutta sen toteuttaminen vaatii jo enemmän ammattitaitoa. Layout-työkalu helpottaa tulevaisuudessa verkkopalvelun kehittämistä, koska sen avulla on helppo lisätä uusia moduulipaikkoja.



Kuva 2. Osa etusivun layoutista.

## 2.4 CSS

CSS eli Cascading style sheets on verkkosivustojen tyyli tiedosto, mikä määrittää sivuston eri elementtien tyyliä. CSS:llä voidaan määrittää esimerkiksi fonttien koot, sekä erilaiset värimääritelmät. Myös elementtien kokoa ja asettelua on mahdollista hallita CSS:llä. (Smith & Negrino 2007, 263.) Elementtien asettelua säädellään täytteillä (padding) ja marginaaleilla (margin). Paddingillä tuodaan lisää tilaa elementin ympärille ja margineilla säädellään elementtien välejä toisiinsa. (Collison, Budd & Moll 2007, 28.)

Elementtien tyylien lisäksi CSS:llä voidaan määrittää, miltä elementit näyttävät eri kokoisilla näytöillä (Kuva 3). Verkkosivuja käytetään yhä enemmän ja enemmän mobiililaitteilla, joten on tärkeää, että verkkopalvelu toimii samalla tavalla erikokoisilla laitteilla ja myös näyttää hyvältä (Wiciak 2014, 4). CSS koodiin on mahdollista upottaa komento @media screen and (max-width) komento, jolla voidaan tehdä poikkeuksia elementin CSS asetuksiin erikokoisilla näytöillä (Wiciak 2014, 11).

```
.sp-megamenu-parent .sp-dropdown li.sp-menu-item >a{
  font-family: sans-serif;
  padding: 3px;
  color: #777;
  font-size: 14px;
}
```

Kuva 3. Valikon linkin CSS-muotoilua.

CSS on tärkeä työkalu, ellei jopa tärkein työkalu, verkkopalvelun responsiivisuuden suunnittelussa. Omasta mielestäni verkkopalvelun CSS:n suunnittelu ja toteuttaminen vie eniten aikaa verkkopalvelun toteuttamisessa. Aikaa kuluu paljon, koska on tärkeää seurata miten CSS koodi käyttäytyy erikokoisilla näytöillä. Se vie paljon aikaa, mutta se tekee verkkopalvelusta tehokkaan. CSS:llä voidaan myös nopeuttaa verkkopalvelun latausaikoja piilottamalla elementtejä esimerkiksi mobiilissa. Kaikkia verkkopalvelun elementtejä ei tarvita mobiilissa, joten ne voidaan piilottaa CSS:llä, jolloin palvelun ei tarvitse ladata ylimääräistä dataa.

## 2.5 Moduulit

Joomlan moduuleilla voidaan tuoda erilaista sisältöä sivustolle, kuten kuvia, artikkeleita ja esimerkiksi päivän sään. Moduulit asetetaan sivupohjassa määritettyihin alueisiin, joita kutsutaan moduulipaikoiksi. Sivupohja määrittää missä sivun osassa moduuleita pystyy näyttämään. Moduulien asetukset ja päälle kytkeminen hoidetaan Joomlan moduulihallinnasta. Moduuleja voidaan käyttää komponenttien sisällön näyttämiseen, mutta komponentit eivät tarvitse moduuleita toimiakseen. Moduulia voidaan käyttää esimerkiksi vain tietyn osan tiedoista näyttämiseen, kuten esimerkiksi 4 uusinta artikkelia. (Repo 2012, 27.)

Moduulin asetuksista pystytään määrittämään, jos moduuli halutaan näyttää vain tietyillä sivuilla tai vain tietyille käyttäjille. Moduulien monipuoliset asetukset mahdollistavat monipuolisen palvelun luomisen Joomlailla. Tarvitaan kuitenkin ammatillista osaamista, jotta moduuleista saa kaiken hyödyn irti. Ammatillista tietoa tarvitaan varsinkin silloin, kun moduulien kanssa syntyy ongelmia, jotka liittyvät käytettävyyteen.

Koska moduuli hakee tietoa jostakin komponentista, voi tulla ongelmia, jos sama komponentti näytetään moduulin kanssa samalla sivulla. Yleensä tämä näkyy siten, että sama tieto näkyy useaan kertaan. Moduulien muokkaamiseen tarvitaan myös CSS:n osaamista, jotta moduuli näyttää hyvältä myös mobiilissa. Mielestäni moduulit mahdollistavat helpon sisällön näyttämisen, mutta jotta si-

sältö näyttäisi hyvältä tarvitaan paljon ammatillista osaamista. Moduuleilla voidaan myös nopeuttaa verkkopalvelun lataamista esimerkiksi määrittelemällä kuvien koot moduulissa. Verkkopalveluihin ladataan paljon erikokoisia kuvia, joten kuvien koot on hyvä tarkistaa moduulin asetuksista. Jos moduuli muuttaa suurta kuvaa pienemmäksi, aiheuttaa tämä kuvan pienennys ylimääräistä latausaikaa. Tästä syystä olisi suositeltavaa tarkistaa, että moduuli käyttäisi oikean kokoista kuvaa.

### **3 Opinnäytetyön lähtökohta ja menetelmälliset valinnat**

#### **3.1 Toimintaympäristö ja lähtötilanne**

Testausta lähdetään toteuttamaan Karjalaisen testiserverillä, jolle on tehty täydellinen kopio Karjalaisen sivustosta. Seuraavaksi aloitetaan layout-työkalulla sivuston templatien rakentaminen. Layoutin valmistuttua kytketään sivuille niille ominaiset moduulit, jonka jälkeen aloitamme niiden CSS-muotoilujen rakentamisen. Muotoilujen rakentamisen yhteydessä testaamme myös moduulien responsiivisuuden ja muokkaamme CSS:ää tarpeen mukaan.

Opinnäytetyön aikana rakennetaan ensin raakaversio nykyisestä Karjalaisen sivustosta, jonka jälkeen aloitamme CSS:n hienosäätämisen. Hiensäätämisen aikana säädetään elementtien ulkoasun vastaamaan täydellisesti Karjalaisen sivustoa. Samalla parannetaan myös ongelmakohtien toimivuutta tarpeen mukaan. Lähestyessä sivuston julkaisua pyydetään vielä palautetta ja korjausehdotuksia sähköisen liiketoiminnan päälliköltä Risto Salmiselta. Korjausten jälkeen julkaistaan uusi framework ja korjataan mahdolliset virheet.

Opinnäytteen toimintaympäristönä oli Pohjois-Karjalan Kirjapaino Oy:n tietohallinto. Opinnäytetyön tekijän lähimpänä esimiehenä toimi tietohallinnon järjestelmäasiantuntija Arttu Romo. Hän vastaa Karjalaisen verkkopalvelun ja muiden sähköisten palveluiden ylläpidosta. Opinnäytteen projektin tilaajana toimi Karjalaisen sähköisen liiketoiminnan päällikkö Risto Salminen. Projektissa työskente-

lee myös Karjalaisen yhteistyökumppani Fiare Oy, joka vastaa Karjalaisen sivuston ylläpidosta. Fiaren tehtäviin kuuluu sivuston komponenttien päivittäminen ja tarvittaessa niiden muokkaaminen. Projektin kohderyhmänä ovat Pohjois-Karjalan maakunnan asukkaat ja erityisesti Sanomalehti Karjalaisen asiakkaat.

Karjalaisen uutta frameworkia varten tehtiin uusi serveri, jossa voisin testata ja muokata materiaalia tarpeen mukaan. Serverille kopioitiin koko pääsivu aineistoinneen, mikä mahdollisti monipuolisen testaamisen, jotta voisin tarttua ongelma-kohtiin heti sellaisen huomattuamme. Kun serveri oli valmis, ryhdyin testaamaan erilaisia Helix3:n layout-asetuksia, jotta saisimme selville, onko Karjalaisen silloista sivua vastaavan templatien tekeminen mahdollista Helix3:lla.

Aluksi kokeilimme templatea, joka olisi jaettu osiin. Tässä sivupohjassa jokaisella osalla olisi ylimääräinen moduulipaikka sivulla oleville moduuleille. Sivupohjan alustavan suunnitelman voi nähdä liitteestä 1. Templatien ylimmät alueet olisi varattu moduuleille ja alemmat K2-komponentille. Templatessa oli kuitenkin se ongelma, että jos ylemmät moduulipaikat olivat tyhjinä, niin niiden sivuilla olevat paikat asettuisivat keskelle aluetta. Testasin useampaa erilaista versiota samanlaisesta asettelusta, mutta idea kaatui aina joko siihen, että moduulialue oli tyhjä tai komponenttialue oli tyhjä. Sivulla olevat moduulit asettuivat väärin aina, kun jompikumpi alueista oli tyhjänä.

Ensimmäisen viikon aikana testasin myös ideaa kahdesta erillisestä templates-ta. Yksi template oli sellainen, jossa käyttäisin ainoastaan moduuleita ja toinen, jossa käyttäisin ainoastaan komponenttia. Testaamisen jälkeen idea vaikutti toimivan hyvin ja templatien prototyypin valmistuttua oli aika tehdä suunnitelma sivun rakenteesta. Tein ensimmäisen viikon aikana rautalankamallin eli wireframen kaikista sivutyypeistä, joita käytettiin Karjalaisen sivulla. Sivutyyppejä oli kaikkiaan kolme kappaletta. Yksi sivu oli etusivu, jossa näytettiin materiaalia erilaisilla moduuleilla keskellä sivua ja sivun reunoilla. Toinen oli kategoria ja tagisivut, joissa listattiin erilaisia artikkeleita sivun keskellä ja sivun reunoilla oli moduuleita. Kolmas sivutyyppi oli yksittäinen artikkeli, missä sisältö näytetään keskellä ja reunoilla moduuleja. Kaikilla sivutyypeillä oli myös header sivun yläosassa ja footer alaosassa (W3 2011a; 2011b). Headerissa näytettiin sivun na-

vigaatio, logo ja mainospaikkoja. Footerista löytyvät linkit Pohjois-Karjalan Kirjapaino Oy:n muihin palveluihin. Sivutyyppeiden wireframet voi nähdä liitteessä 2. Onnistuimme myös löytämään Helix3:n omilta kotisivuilta ohjeen, jolla voisimme rakentaa kustomoidut moduulipaikat komponentin templateen. Tällä tavoin voisimme kytkeä myös moduuleita näkymään samalla alueella, jossa näytettiin komponenttia. Ohjeistimme Fiarea lisäämään kyseiset moduulipaikat Helix3:n asetuksiin.

Rautalankamallin valmistuttua pystyimme päättämään, että tarvitsimme moduulialueella toimivaa sivupohjaa etusivulla ja muut sivut käyttäisivät komponentilla toimivaa sivupohjaa. Pohdin tällöin vielä sitä, että vaikuttaisiko kahden templatien sivusto sivun latausaikaan. Kysyin asiasta järjestelmäasiantuntijan mielipidettä ja hänen mielestään olisi hyvä, jos voisimme joskus jättää komponenttialueet kokonaan pois käytöstä. Tämä helpottaisi paljon, koska silloin ei olisi ongelmaa, jossa moduuli näyttää saman uutisen kuin komponenttikin. Päätimme kuitenkin nyt tämän projektin aikana jatkaa kahden templatien idealla, koska se vastasi silloista Karjalaisen sivustoa eniten. Moduuli- ja komponenttialueiden lisääminen onnistuu helposti Helix3-frameworkin layout-työkalulla, mikä helpottaa verkkopalvelun kehittämistä tulevaisuudessa.

### **3.2 Toiminnan eteneminen**

Ensimmäisellä viikolla tehdyn wireframen avulla oli helppo aloittaa sivuston tekeminen. Toisella viikolla kytkin sivuston moduuleita käyttöön, jotta sivustolle saatiin sisältöä. Sivustolla oli jo valmiiksi samat moduulit, joita käytettiin silloisella Karjalaisen sivustolla. Kuitenkin moduuleilta puuttui kaikki CSS-muotoilut, joten sivusto oli hyvin sekavan näköinen. Keskusteltuani järjestelmäasiantuntijan kanssa päätimme että, oli parasta aloittaa työstämään sivustoa ylhäältä alaspäin ja aina kun yksi elementti olisi valmis tietokoneen näytöllä tekisimme siitä responsiivisen. CSS:n suunnittelemiseen käytin Google Chromen Devtoolsin Elements-valikkoa, joka näyttää verkkopalvelun rakenteen. Elements valikon saa auki klikkaamalla elementtiä hiiren vasenta painiketta ja klikkaamalla "inspect". Elements-valikosta näkee esimerkiksi moduulien class-nimet, joita voi käyttää moduulien tyylittelyssä.

Rakentelin moduulin muotoilut ja responsiivisuuden etusivulla valmiiksi, minkä jälkeen se oli helppo kopioida muiden sivujen templatien custom CSS osioon. Etusivulla oli paljon moduuleja, joita käytettiin myös muillakin sivuilla, joten oli helpompaa rakentaa etusivulla kaikki valmiiksi ennenkuin siirryin muiden sivujen pariin. Yleisiä asetuksia kuten tekstin ja linkkien fonttiasetukset tein myös aluksi etusivulla ja kopioin ne sen jälkeen muille sivuille. CSS:ää kirjoittaessa on tärkeää pysyä mahdollisimman järjestelmällisenä. Järjestelmällisyys helpottaa koodin kirjoittamista huomattavasti, koska ongelman sattuessa tiesin heti mistä etsiä ratkaisua. Kirjoitin ensimmäiseksi CSS:ään moduulipaikkojen muotoilut, jonka jälkeen seuraavana olivat tekstien muotoilut. Seuraavaksi lisäsin moduulien muotoilut, jossa ylimpien moduulien muotoilut olivat ensimmäisenä ja alimpien viimeisenä. Samankaltaista järjestystä käytin myös responsiivisuutta suunnitellessani, kuten esimerkiksi media queryjä kirjoittaessani, jossa kirjoitin koodin suurimmasta näyttökoosta pienimpään. Verkkopalvelun latausnopeus laskisi hieman, kun CSS:ää ladattaisiin kahdesta eri paikasta, mutta julkaisun jälkeen CSS koottaisiin yhteen tiedostoon. Selkeä järjestys helpottaisi kahden erillisen tiedoston yhdistämistä ja yhdistetty tiedosto lopulta koko sivuston latausnopeutta.

### 3.3 Navigaatio ja layout

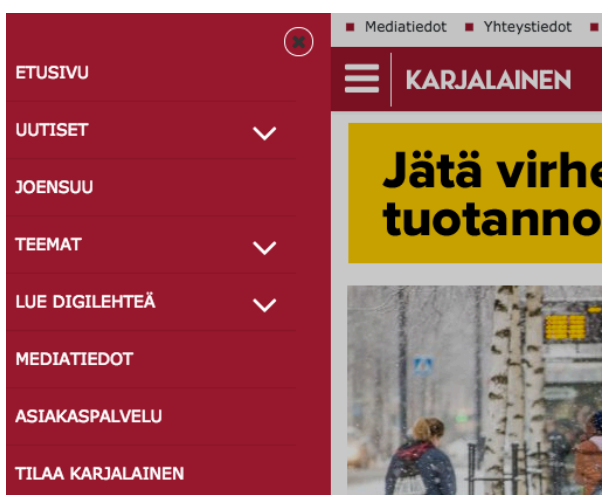
Sivuston navigaatiovalikko oli jo hieman tyyllitelty, koska projektia oli yritetty aloittaa jo aiemmin. Navigaation painikkeissa ei ollut korjattavaa, mutta muut elementit navigaatioissa tarvitsivat tyyllittelyä (Kuva 4). Navigaation oikeaan reunaan tulee "Etsi" -moduuli, "Tilaa Karjalainen ja Asiakaspalvelu" -painikkeet ja ikoni, josta käyttäjä voi kirjautua sivulle. Asetin elementeille float right-arvon, jotta saisin ne asettumaan oikeaan reunaan. Elementit kuitenkin asettuivat väärin päin, joten jouduin säätelemään niiden margineita ja paddingejä, jotta se vastasi silloista Karjalaisen sivua. Tämän jälkeen jouduin asettelemaan elementeille tarkat pikselileveydet, koska jos annoin elementeille prosenttileveydet, ne eivät pysyneet paikallaan sivua skaalatessa. Mobiilissa "Tilaa Karjalainen" ja "Asiakaspalvelu"- painikkeita ei tarvittu, joten piilotin ne `display: none;`-komennolla CSS:ssä.





Kuva 4. Navigaatio ja sen moduulit.

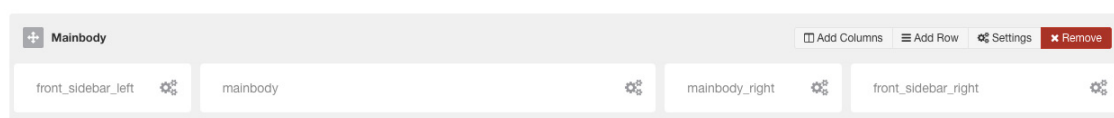
Helix3:ssa oli myös tavallisen navigaatiovalikon lisäksi "offcanvas"-valikko, jota käytetään mobiililaitteilla navigaatiovalikon sijaan (Kuva 5). Mobiilivalikko toimi oikein hyvin eikä tarvinnut suuria muutoksia. Mobiilivalikon paikkaa piti kuitenkin alustavasti muokata, koska Karjalaisen sivulla mobiilivalikko oli sivuston vasemmassa reunassa. Löysin ongelmaan ratkaisun Helix3:sen kehittäjän Joomshaperin foorumeilta (Toufiqur 2017). Ongelman ratkaistunani korjasin myös mobiilimenun nuolia, joilla kuvataan että valikolla on alavalikko. Nuoli oli mielestäni liian pieni, joten suurensin sitä helpottaakseni valikon käyttöä mobiilissa. Uudella mobiilivalikolla onnistuttiin parantamaan Karjalaisen verkkopalvelun mobiilikäytettävyyttä, koska vanha mobiilivalikko oli vanhanaikainen, sekä hankala muokata. Uuteen mobiilivalikkoon on helpompi lisätä uusia ominaisuuksia, koska siihen voi kytkeä moduuleita. Painikkeet "Mediatiedot", "Asiakaspalvelu" ja "Tilaa Karjalainen" lisättiin käyttämällä moduulia, jotka lopuksi tyylieltiin CSS:llä. Mobiilivalikko löytyy "offcanvas" nimellä myös moduulipaikoista, joten tulevaisuudessa on mahdollista lisätä erilaisia ominaisuuksia myös mobiilikäyttäjille.



Kuva 5. "Offcanvas"-valikko.

Menun jälkeen siirryin muokkaamaan frameworkin layout-työkalulla etusivun rakennetta, jota olin rakennellut layout-työkalulla jo aiemmin testausvaiheessa, mutta jouduin nyt syventymään siihen tarkemmin. Mittailin aluksi silmämääräi-

sesti, miltä sivuston tulisi näyttää. Lopulta sain tulokseksi, että sivuston layout kolumnit tulee jakaa 2,5,2,3, jotta saataisiin samanlainen asettelu kuin Karjalaisen silloisella sivustolla. Etusivulle loin kärkiuutisia varten oman moduulialueen, jonka nimeksi tuli mainbody\_right (Kuva 6). Muut etusivun moduulit asetin mainbody-paikkaan. Etusivulla mainbodyn leveys oli 40 % alueesta ja mainbody\_right alueen leveys oli 19.4 %. Sivulla olevien alueiden leveys oli 15 % ja 25 %. Kaikki alueet veivät siis yhteensä 99,4 % 1200 pikselin leveydestä. Jouduin jättämään 0,6 % ylimääräistä, koska usein sivuston layout meni rikki, jos käytin liian tarkkoja mittoja ja tarvitsin myös tilaa margineille ja paddingeille.



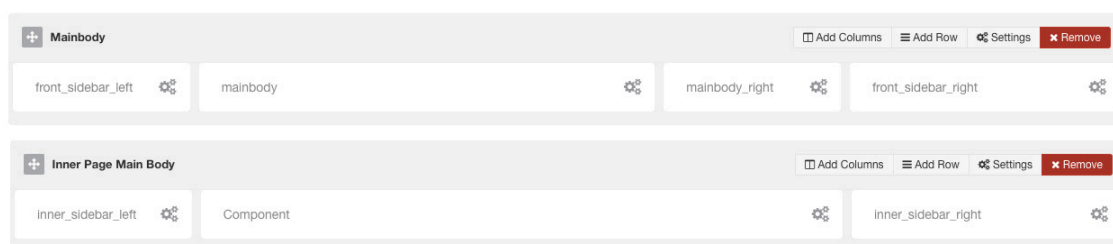
Kuva 6. Etusivun layout.

### 3.4 Pääuutinen ja kärkiuutiset

Moduulit “Pääuutinen” ja “Kärkiuutiset” olivat menun jälkeen seuraavia tyylliteltäviä asioita. Kun sain moduulit näkymään paikoillaan aloitin niiden leveyksien säätämisen. Mainbodyn 40 %:n leveyden takia jouduin kasvattamaan joidenkin moduulien leveyksiä yli 100 %. “Pääuutinen”-moduuli esimerkiksi on 100 %:n leveydessä mainbodyssa, mutta kaikki muut moduulit ovat 154 %:n levyisiä, jotta ne vastasivat silloista Karjalaista. Kärkiuutiset oli helppo muotoilla mainbody\_right-alueeseen sopivaksi, mutta ongelmia syntyi kuitenkin mobiilikokoon skaalatessa. Moduuli siirtyi mobiiliin skaalatessa jatkuvasti eri kohtaan, mikä johtui skaalautuvasta sisällöstä. Karjalaisen sivulla kärkiuutiset olisivat vielä jatkaneet pääuutisen vieressä noin 80 pikseliä, jonka jälkeen ne siirtyivät pääuutisen alle mobiilissa. Uuteen frameworkiin kuitenkin tein media queryn, jossa kärkiuutiset menevät aiemmin pääuutisen alle. En kuitenkaan saanut moduulia asettumaan suoraan pääuutisten alle, koska se oli erillisessä moduulipaikassa. Löysin kuitenkin ratkaisun käyttämällä {loadposition} ominaisuutta, jonka avulla pystyin lataamaan moduulialueen toisen alueen sisälle (Joomla 2016). Rakensin tätä varten custom html-moduulin, johon upotin {loadposition}-komennon. Olisin myös voinut käyttää {loadmodule}-ominaisuutta, mutta en saanut sitä jostain syystä toimimaan. Idea näytti toimivan kuitenkin oikein ja alueella ei käytetä

muuta moduuleita, joten idea olisi toiminut kummalla tahansa tyyllillä. Tämän jälkeen tarvitsi piilottaa CSS:llä mainbody\_right ja asettaa korvaava moduuli näkymään display: block-komennolla.

Muutkin sivut käyttävät ”Pääuutinen” ja ”Kärkiuutiset”-moduuleita, mutta niissä käytetään erilaista moduulia kuin etusivulla. Etusivun moduulit käyttävät K2 Content-moduulia, johon käyttäjä voi itse valita artikkelit. Muilla sivuilla käytetään GK5 News Show Pro-moduulia, joka voidaan asettaa hakemaan uusin artikkeli kategoriasta tai tagista. Olisin aluksi halunnut käyttää kaikkialla samoja moduuleita, mutta tarvitsimme ominaisuutta, jossa käyttäjä voi valita artikkelit eikä sellaista ominaisuutta ollut GK News Show:ssa. Jos olisimme käyttäneet samoja moduuleita, olisimme voineet nimetä moduulien classit samanlaisiksi. Tällä tavalla olisimme voineet tehdä yhden toimivan koodin, jonka sitten kopioisin muille sivuille. Muiden sivujen kärkiuutisista puuttuivat kuvat, mutta muuten muotoilut muistuttavat etusivun tyyliä. Muilla sivuilla oli kuitenkin ongelmana se, että moduuleita ei saanut komponenttialueeseen kytkettyä (Kuva 7). Onnistuimme testausvaiheessa kuitenkin löytämään ratkaisun Joomshaperin kotisivulta, jossa kerrottiin miten komponentti alueeseen saatiin ”above” ja ”below” moduulipaikat (Joomshaper 2015).



Kuva 7. Etusivun ja muiden sivujen layoutin erot. Kustomoidut moduulipaikat eivät näy layoutissa, vaikka ovatkin olemassa.

Kunhan Fiare sai moduulipaikat lisättyä, ryhdyin testaamaan moduuleja. Moduulit eivät kuitenkaan toimineet oikein, vaan useimmiten ne menivät toistensa kanssa päällekkäin. Yritin säädellä moduuleitten järjestystä ja float arvoja, mutta mikään ei toiminut tahtomallani tavalla. Testatessani moduulipaikkaa heräsi kuitenkin ajatus, jos olisi mahdollista jakaa ”above”-moduulipaikka kahteen erilliseen moduulipaikkaan. Asiasta tehtiin selvityspyyntö Fiarelle ja he onnistuivat

jakamaan alueen, joten saimme komponenttialueelle yhteensä kolme moduulipaikkaa “above-left”, “above-right” ja “below”. Korjausten jälkeen kytkin pääuutisen left paikkaan ja kärkiuutiset right paikkaan ja moduulit toimivat haluamallani tavalla. Moduulien leveyksiä piti kuitenkin säätää CSS:llä, koska “above” oli jaettu kahteen 50/50 osioon, joten molemmat moduulit olivat alustavasti saman levyisiä. Jouduin venyttämään CSS:llä pääuutisia hieman kärkiuutisten päälle, jotta sain samanlaisen tuloksen kuin Karjalaisen sivulla.

Pääuutiset ja kärkiuutiset moduuleita jouduin muokkaamaan vielä useaan otteeseen ennen julkaisua ja myös julkaisun jälkeen tein vielä korjauksia. Muiden sivujen kärkiuutisiin tein ennen julkaisua tarkat pikselileveydet, jotta sain ne toimimaan oikealla tavalla mobiiliin skaalatessa. Mielestäni pikselileveyksien käyttäminen on hieman vanhanaikaista, koska laitteet kehittyvät jatkuvasti ja uusia näyttökokoja ilmestyy nopeaan tahtiin. Etusivun kärkiuutiset aiheuttivat vielä julkaisun jälkeenkin ongelmia, koska ne ovat eri moduulipaikassa. Karjalaisen verkkopalvelua kehitetään kuitenkin jatkuvasti, joten ongelmaan ollaan jo suunnittelemassa ratkaisua.

### **3.5 Class-arvot ja kuvien optimointi**

Palvelun muissa moduuleissa pyrin käyttämään mahdollisimman paljon samoja class-arvoja samanlaisissa moduuleissa. Etusivulla olevat “Sää Joensuussa”, “Uusimmat uutiset”, “Uusimmat maakunnasta” ja “Osallistu” –moduulit käyttävät samaa light nimistä class-arvoa. Tämä helpotti muotoilua, koska tiesin moduulien näyttävän alustavasti samanlaisilta ja käyttävän samoja fontteja. Moduuleissa tarvittiin kuitenkin erikokoisia fontteja, kuten esimerkiksi “Osallistu” –moduulissa. Tällaisissa tapauksissa yleensä lisäsin class-nimeen uuden nimen lisäksi, joten “Osallistu” –moduulin class nimeksi tuli ”light osallistu”. Tällöin moduuli säilytti muotoilussaan light classin arvot, mutta pystyin myös lisäämään erilaisia muotoiluja kyseiseen moduuliin muokkaamatta muita light classin moduuleita. Samanlaista käytäntöä on käytetty etusivulla olevien “Maakunta”, “Kulttuuri”, “Urheilu” ja “Autot” –moduulien otsikkojen muotoiluissa.

Class-arvoja käytettiin myös kategoria- ja tagilista sivuilla, joissa class arvoja käytettiin erottamaan kuvalliset artikkelit kuvattomista artikkeleista. Jouduimme

lisäämään classit listasivuille, koska kuvallisten artikkeleiden tekstit menivät silloin tällöin kuvien alle. Yritimme alustavasti hallita tekstiä säätelemällä kuvan margin-bottom arvoja, mutta siltikin osalla artikkeleista teksti meni kuvan alle. Sama ongelma ilmeni myös ennen julkaisua kategorialistasivuilla, mutta classit saatiin julki vasta julkaisun jälkeen. Samankaltaisia class-arvoja tarvittiin myös moduuleihin. Class-arvojen lisäämiseksi tarvitsimme yhteistyökumppaniamme Fiarea, mutta responsiivisuuden ja muotoilun suunnittelimme itse. Jatkossa näiden class-arvojen avulla on helpompaa luoda uusia moduuleita, kun samoja ominaisuuksia tarvitaan jatkossakin. Käyttämällä samanlaisia class-arvoja voimme helposti kopioida jo valmiista moduuleista tiettyjä ominaisuuksia, joten kaikkia muotoiluja ei tarvitse tehdä alusta asti.

Karjalaisen verkkopalvelussa käytetään paljon erikokoisia kuvia, joten oli tärkeää tutkia millaisia kuvia käytetään eri tilanteissa. Keskustelimme aluksi järjestelmäasiantuntija Romon kanssa ja tulimme tulokseen, että pienimmillä näytöillä suurin osa kuvista voitaisiin piilottaa. Tämä nopeuttaisi palvelun latausta mobiililaitteilla, eikä kuville tarvitsisi rakentaa muotoiluja. Kuvien kokoja pyrittiin myös parantamaan siten, että kuvien kokoa ei muutettaisi CSS:llä liikaa. Jos kuvia skaalattaisiin pienestä isommaksi, tai isosta pienemmäksi, aiheuttaisi tämä ylimääräistä dataliikennettä, joka taas hidastaisi latausnopeutta. Ensin tutkin elements-valikon avulla kuvien kokoja, mitkä sopisivat parhaiten tekemääni ulkoasuun. Tämän jälkeen kuvien kokoja säädettiin K2-komponentin asetuksista, jossa määritellään ladattavan kuvan koko ja kuvan laatu. Karjalaisen vanhoilla asetuksilla pienten kuvien koko oli 250 pikseliä, mutta osalla listasivujen kuvissa koko oli n. 200 pikseliä. Tämä suurennos aiheutti ylimääräistä latausaikaa, joten asetin pienten kuvien kooksi 200 pikseliä. Tein samanlaisia säätöjä myös suuriin ja keskikokoisiin kuviin. Verkkopalveluun tulee kuvia myös muilta tarjoajilta, kuten STT:ltä ja mainospalveluista, joiden kuvissa saattaa tapahtua skaalautumista. Kuitenkin Karjalaisen omat kuvat toimivat ilman skaalausta, joten verkkopalvelun latausnopeutta saatiin lyhennettyä.

### **3.6 Menun käyttäminen tabletilla**

Palvelun testausvaiheessa huomasin, että palvelun navigaation käyttäminen tabletilla oli hankalaa. Navigaatiossa olevat alavalikot eivät aina auenneet vaan

useimmiten valikon painaminen avasi päävalikon linkin. Tabletti käytti kuitenkin pystyasennossa mobiilimenua, joka toimi oikein. Vaaka-asennossa tabletti käytti samaa navigaatiota kuin tietokoneen näytöllä. Pyysimme aluksi Fiarea tutki-  
maan, voisiko menusta tehdä sellaisen, että kerran painamalla alavalikko au-  
keaisi ja toisella painalluksella aukeaisi päävalikko. Tämänkaltainen ratkaisu oli  
käytössä Karjalaisen sivustolla, mutta Helix3:sen navigaatioon ei kuitenkaan  
saatu samankaltaista ominaisuutta rakennettua. Ryhdyin tutki-  
maan, olisiko mo-  
biilimenun mahdollista kytkeä päälle tabletin ollessa vaaka-asennossa. Löysin  
 netistä paljon tietoa erilaisista media queryistä, joilla pystyttiin säätelemään  
elementtejä riippuen laitteen ruudun asennosta (Gilbert 2017). Asiaa tutkiessa  
ja testatessa meni pitkään, enkä saanut koodia toimimaan haluamallani tavalla.  
Alustavasti oletin että tarvitaan asetus sekä pystyasennolle että vaaka-  
asennolle. Aikaa kului, kun yritin lisätä koodia kahteen uuteen media queryyn ja  
saada ne toimimaan siten etteivät ne vaikuta muihin queryihin.

Uusi framework oli kuitenkin niin lähellä valmista, että päätimme jättää ongel-  
man myöhemmin ratkaistavaksi. Julkaisun jälkeen palasin asiaan, kunhan  
olimme julkaisun jälkeen tehneet tarvittavat korjaukset. Pääsin testaamaan me-  
dia queryja, kunhan uusi testipalvelin oli valmis. Ongelma ratkesi lopulta siten,  
että jätin kokonaan pystyasento-asetukset pois media queryistä ja asetin vaaka-  
asento media querylle tarkat laiteleveydet. Media queryn kirjoitin seuraavasti  
`@media only screen and (min-device-width: 768px) and (max-device-width:  
1024px) and (orientation : landscape)`. Media queryn sisään kirjoitin koodin, jos-  
sa asetin mobiilimenu painikkeen ja mobiilimenun näkyviin, sekä piilotin työpöy-  
dällä näkyvän navigaation. Testasin koodia monella eri tabletilla ennen koodin  
kopiointia viralliselle sivustolle. Verkkopalvelun käytettävyyttä saatiin paran-  
nettua, kun kaikilla mobiililaitteilla on samanlainen valikko käytössä riippumatta  
laitteen suunnasta tai koosta.

### **3.7 Custom CSS-moduulit ja adblock**

Osalla sivuista tarvittiin myöhemmin hieman erilaisia muotoiluja linkkeihin ja  
moduuleihin. Esimerkiksi asiakaspalvelu sivulle haluttiin linkin fontille punainen  
väri, joka muuttuu mustaksi hiiren mennessä linkin päälle. Tavallisesti linkki on  
musta, joka muuttuu punaiseksi hiiren mennessä päälle. Nämä muotoilut olisi-

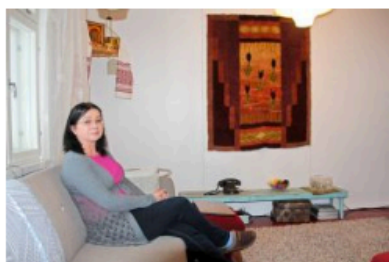
vat voineet vaikuttaa muihinkin elementteihin, joten oli yleensä helpompaa tehdä tämänkaltaisille sivuille custom CSS-moduuli. Custom CSS-moduulilla voidaan lisätä tavallisten muotoilujen lisäksi myös uutta CSS-koodia. Moduulin on myös hyödyllinen, koska se voidaan asettaa tietyille sivuille, joten se ei vaikuta kaikkiin sivuilla oleviin elementteihin, vaan ainoastaan niihin elementteihin, jotka ovat kohdistetulla sivulla. Custom CSS-moduulia käytettiin esimerkiksi kunta-vaalien sivujen ulkoasun tekemiseen sekä yksittäisten artikkelisivujen muotoiluun, kuten esimerkiksi ”Asiakaspalvelu”-sivun linkkeihin.

Testasin sivuston toimivuutta myös ilman mainoksia. Sivustolla ei ollut muuten ongelmia, mutta etusivun rakenne meni sekaisin, jos adblock oli käytössä. Rakenne rikkoutui pääuutisen alla olevan mainospaikan takia. Jos mainosta ei näytetty pääuutinen ja kärkiuutiset moduuli menivät toistensa päälle. Halusin korjata ongelman, vaikka adblockin käyttöä ei pitäisi suosia. Onnistuin ratkaisemaan ongelman siten, että loin tyhjän custom HTML-moduulin ja asetin sen mainosmoduulin päälle. Tämän jälkeen asetin mainosmoduulille `position: absolute;`-komennon CSS:ssä, jolloin moduuli ei reagoi lähellä oleviin elementteihin. Tämän jälkeen asetin sen z-indeksin 1 ja tyhjän moduulin z-indeksiksi 0, jolloin mainosmoduuli meni tyhjän alueen päälle. Tämän jälkeen jos käyttäjä tulisi sivustolle adblock päällä mainos häviäisi, mutta tyhjä alue jäisi mainoksen tilalle. Suunnittelin moduulia aluksi paperille, joka on liitteessä 3.

### **3.8 Julkaisu**

Ennen julkaisua olimme saaneet Helix3:lla tehdyn frameworkin vastaamaan Karjalaisen verkkopalvelua, mutta muutamia ongelmia oli kuitenkin vielä navigaation käyttämisessä tabletilla sekä kategorioiden teksteissä. Ongelmista huolimatta päätimme julkaista sivun, koska ongelmat voitaisiin ratkaista julkaisun jälkeenkin (Kuva 8).

maakaupasta ja sukupolven perustavasta syntyisestä.



## Kaukaan herrojen asunnosta tuli taidetta rakastavan pariskunnan persoonallinen koti

MAAKUNTA 3.2.2017 20:43 TILAAJILLE

Jos Kiteen Puhoksessa sijaitseva entinen Oy Kaukas Ab:n talo osaisi puhua, sillä olisi monta mielenkiintoista tarinaa kerrottavanaan. Sen jyvien hirsiseiniin suojassa on asunut niin herroja kuin työläisiäkin. Yhtiö luopui kiinteistön

omistuksesta 1980-luvun puolivälissä, ja talo siirtyi Kiteen kunnan omistukseen Puhoksen ranta-alueita koskevan maakaupan yhteydessä.

Kuva 8. Kuva kategorian teksteistä ennen korjausta.

Uuden frameworkin julkaiseminen ei sujunut ongelmitta, koska julkaisun aikana jokin meni vikaan ja osa sivupohjan asetuksista katosi. Ongelma saatiin korjattua seuraavina päivinä, mutta ongelman syy jäi mysteeriksi. Julkaisun jälkeen Trellosta saamamme palautteen avulla saimme myös korjattua pienet yksityiskohdat, jotka olivat testatessa jääneet huomaamatta. Suurimmat ongelmat kuitenkin, jotka saatiin ratkaistua julkaisun jälkeen olivat navigaation käyttö tablettissa ja kategorioiden tekstiongelmät.

### 3.9 Yhteenveto prosessista

Opinnäytetyöni tekemiseen meni noin neljä ja puoli kuukautta. Näihin neljään ja puoleen kuukauteen mahtui paljon testausta, hienosäätöä ja alusta aloittamista. Olen koonnut näihin edellisiin lukuihin mielestäni tärkeimmät osuudet Karjalaisen verkkopalvelun frameworkin uudistuksesta. Frameworkin uudistamisen aikana tarvittiin paljon fonttien, responsiivisuuden, marginien, paddingien ja erilaisten elementtien hienosäätöä, joiden parissa saattoi kulua jopa päiviä. Prosessin aikana ajattelin usein jonkin elementin olevan valmis, mutta jouduin kuitenkin palaamaan sen pariin useaan kertaan.

Opinnäytetyötä tehdessä rakensin ensin kopion jostakin elementistä ja tyyllittelin sen mahdollisimman toimivaksi eri laitteissa. Elementin valmistuttua raportoin järjestelmäasiantuntijalle joko skypeillä tai suoraan kertomalla, jonka jälkeen hän testasi elementtiä itse. Joidenkin elementtien kanssa tarvitsimme Fiaren apua, koska kaikkia ongelmia ei pystytty ratkaisemaan pelkästään asetuksilla tai



CSS:llä. Tällöin teimme asiasta ilmoituksen Redmine projektinhallintaohjelmaan, jota käytämme Fiaren kanssa ongelmien raportoimiseen. Ongelman laajuudesta riippuen vika saatiin korjattua päivän tai kahden aikana. Opinnäytetyön aikana oli kuitenkin useita eri projekteja käynnissä, joten joskus ongelman ratkaisua sai odottaa pidempäänkin. Hieman ennen julkaisua pyysimme myös Risto Salmisen mielipidettä uudesta frameworkista ja pyysimme myös korjausehdotuksia. Julkaisun jälkeen otimme käyttöön Trello projektinhallintaohjelmiston, johon kutsuimme myös verkkotoimituksen työntekijöitä. Trello helpotti julkaisun jälkeisten ongelmien ratkaisemisessa, kun saimme palautetta verkkotoimittajilta reaaliajassa sivuston käyttäjiltä.

Opinnäytetyön tekeminen sujui mielestäni saumattomasti. Työlle ei asetettu virallista deadlinea, mikä rentoutti työskentelytahtia. Myös työskentelytavat helpottivat työn etenemistä. Pystyin siirtymään seuraavan elementin tyyllittelyyn, kun edellistä testattiin. Mikäli emme löytäneet itse ratkaisua ongelmaan, teimme asiasta selvityspyynnön Fiarelle. Julkaisun jälkeen avattu Trello helpotti myös työn etenemistä, koska työn valmistuttua pystyi ohjelman kautta pyytämään tarkistusta sisällöntuottajilta reaaliajassa.

Kokonaisuudessaan projektin sisäisessä toiminnassa ei ollut ongelmia. Aloitimme projektin ensin pienessä piirissä, jossa kysyin ensin palautetta pelkäästään järjestelmäasiantuntija Romolta. Lähestyessämme valmista tuotetta pyysimme palautetta myös Risto Salmiselta. Palautetta tuli fonttien tyyleistä sekä muutaman moduulin asetuksista. Ongelmat kuitenkin saatiin ratkaistua pienillä CSS:n korjauksilla sekä moduulien asetuksia korjaamalla. Saamiemme muutosehdotusten jälkeen julkaisimme sivun, jonka jälkeen saimme palautetta myös palvelua käyttäviltä työntekijöiltä.

## 4 Tulokset

Opinnäytetyön yhtenä lähtökohtana oli Karjalaisen verkkopalvelun latausaikojen lyhentäminen. Latausajat olivat aluksi lähempänä viittä sekuntia vanhalla sivupohjalla. Julkaisun jälkeen testasin palvelun latausaikaa ja sain tulokseksi noin

kolme sekuntia. Käytin nopeuden testaamiseen pingdom nimistä verkkopalvelua. Sivuston latausaikaa pyritään kuitenkin vielä jatkossa saamaan lyhyemmäksi vähentämällä mainosten aiheuttamaa dataliikennettä. Latausajasta saatiin kuitenkin pelkällä frameworkin päivittämisellä poistettua noin kaksi sekuntia, joten näkisin projektin onnistuneen tällä alueella.

Uuden frameworkin myötä Karjalaisen verkkopalvelua on helpompaa kehittää jatkossa. Frameworkin layout työkalu mahdollistaa uusien moduulipaikkojen luomisen helposti. Uusi framework mahdollistaa myös sivuston rakenteen helpon muokkaamisen tulevaisuudessa, kun sivustoa joskus halutaan uudistaa. Sivuston käyttöä mobiililaitteilla on myös parannettu käyttämällä frameworkin omaa mobiilivalikkoa, joka on aiempaan mobiilivalikkoon verrattuna huomattavasti käyttäjäystävällisempi (Kuva 9).



Kuva 9. Uusi ja vanha mobiilivalikko.

## 5 Pohdinta

Opinnäytetyössä saavutettiin haluamamme päämäärät. Onnistuimme laskemaan sivuston latausaikaa, jopa kahdella sekunnilla. Latausajan pienentämisen lisäksi frameworkin layout työkalu avaa uusia mahdollisuuksia Karjalaisen verkkopalvelun kehittämiseen tulevaisuudessa. Opinnäytetyön toteuttamiseen käy-

tettiin noin neljä ja puoli kuukautta. Työ olisi ehkä valmistunut nopeammin, jos olisi ainoastaan keskitytty frameworkin uudistamiseen. Karjalaisen tietohallinnossa on kuitenkin useita projekteja käynnissä yhtä aikaa, joten tärkeämmät työt menivät frameworkin edelle. Tuolloin olivat muun muassa kuntavaalit sekä Karjalaisen lintuvisa, jotka veivät resursseja frameworkin päivityksestä. Projekti valmistui kuitenkin omasta mielestäni yllättävän nopeasti, kun projekti oli alunperin ollut noin vuoden ”jäissä”. Projektin aikana ei myöskään omasta mielestäni ollut monta sellaista ongelmaa, jonka ratkaisemiseen olisi kulunut kohtuuttomasti aikaa.

Opinnäytetyön aikana opin paljon uutta CSS:stä, Joomlaista, sekä Joomlaan moduulien toiminnasta. CSS:stä itselläni oli ennen tätä projektia melko paljon tietoa, mutta näin suurta projektia en ollut aikaisemmin tehnyt. Sain myös tarvittaessa apua Arttu Romolta ja Risto Salmiselta, jos en tiennyt miten kyseisen moduulin tulisi toimia sivustolla. Neuvojen avulla pystyin muokkaamaan verkkopalvelun moduuleita toimimaan halutulla tavalla. Työn aikana sain kuitenkin melko vapaasti käyttää omaa harkintaani elementtien muotoilujen suunnittelussa. Yleensä etsin verkosta esimerkkejä siitä, miten elementtejä kannattaisi lähteä tyyllittelemään. Löytäessäni sopivan tyylin miten elementin voisi tyyllitellä, rikoin koodin osiin ja kokosin siitä palveluun sopivan muotoilun. Joskus pystyin kuitenkin kopioimaan koodin suoraan vanhasta palvelusta, mutta useimmiten jouduin muokkaamaan koodia Helix3:een sopivaksi.

Omasta mielestäni yksikään projekti ei ole samanlainen. Jos yksi asia toimi toisessa projektissa hyvin, se ei tarkoita sitä, että se onnistuisi yhtä helposti toisessakin. Koodia joutuu usein hajottamaan osiin ja kokoamaan uudestaan. Suurimman osan CSS:stä olen oppinut yksinkertaisesti testaamalla kymmentä eri tapaa löytääkseni yhden toimivan. Tietoperustani perustuu myös paljon suulliseen tietoon. Suurin osa CSS:tä ja muustakin Joomlaan liittyvästä on tarttunut vuosien varrelta niin opettajilta kuin muilta opiskelijoiltakin. Tähän projektiin oli tästä syystä hankalaa löytää lähdemateriaalia, koska CSS:ää ei opi kuin kokeilemalla ja aloittamalla uudestaan.

Ennen projektin päättämistä paljastui, että Google on merkitsemässä HTTP-sivustot suojaamattomaksi, joten jatkossa tulisi käyttää HTTPS-sivustoa

(Schechter 2016). Tämä ilmestyi vähän ennen sivuston julkaisemista, mikä vaikutti myös tekemääni koodiin. Järjestelmäasiantuntijan kanssa keskusteltuani emme voineet sanoa varmaksi, toimisiko tekemäni CSS HTTPS-sivustolla. Tästä syystä emme siirtäneet CSS:ää yhteen tiedostoon ja asettaneet sitä sivun head osioon, kuten suositeltaisiin. Jätimme koodin vielä Helix3:sen custom CSS-osioon, jossa sitä on helppo muokata tarpeen vaatiessa. Sivuston latausaikaa saisi hieman pienennettyä, jos CSS:n kopioisi yhteen tiedostoon ja asettaisi sen yhdelle riville, joten sivua ladattaessa ei selaimen tarvitse lukea koodia rivi kerrallaan.

Verkkopalvelun latausaikaa saisi myös siten nopeutettua, että vähentäisimme mainospaikkojen aiheuttamaa dataliikennettä. Tähän ongelmaan ollaan tällä hetkellä etsimässä ratkaisua Pohjois-Karjalan Kirjapaino Oy:n tietohallinnossa. Jos mahdollista haluaisin korjata etusivulla olevan ”Pääuutinen” ja ”Kärkiuutiset”-moduulien ongelman, jossa kärkiuutiset menevät alla olevien moduulien päälle. Ongelmaa yritin ratkaista ennen projektin päättämistä, mutta lopulta tulimme siihen tulokseen, että olimme saavuttaneet päämäärämme. Ongelmaan ollaan myös kehittämässä ratkaisua tällä hetkellä, mutta alustava tarkoitus olisi lisätä uusi moduuli ”Pääuutiset”-moduulin alle. Moduulin ansiosta ”Kärkiuutiset” saisi lisää liikkumatilaa, mikä ratkaisisi ongelman.

## Lähteet

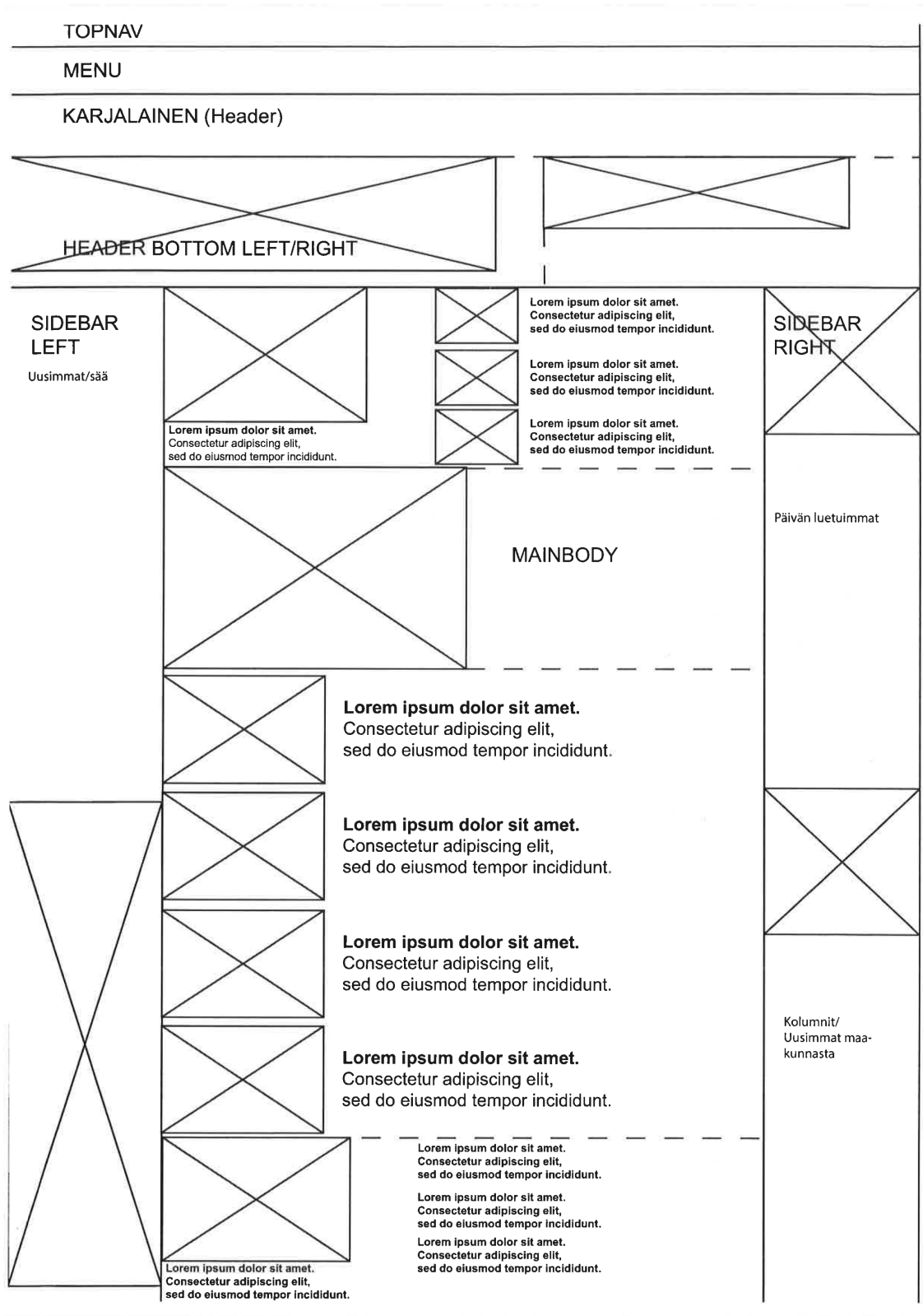
- Blomberg, M. 2015. Siirtyminen responsiiviseen verkkosivusuunnitteluun. Lahden Ammattikorkeakoulu. Tietotekniikan koulutusohjelma. Opinnäytetyö. <http://urn.fi/URN:NBN:fi:amk-201504295520>. 4.2.2017.
- Collison, S., Budd, A., Moll, C. 2007. CSS – Tehokas hallinta. Helsinki: Readme.fi
- Docs.Joomla.org. 2014. Getting Started with Templates. [https://docs.joomla.org/Getting\\_Started\\_with\\_Templates](https://docs.joomla.org/Getting_Started_with_Templates). 15.5.2017.
- Gilbert, S. 2017. CSS Media Queries for Ipad & Iphones. Stephen Gilbert. 23.2.2015. <http://stephen.io/mediaqueries/>. 3.4.2017.
- Joomla.fi. 2017. Mikä Joomla? <http://www.joomla.fi/>. 30.1.2017.
- Joomla.org. 2016. How do you put a module inside an article? [https://docs.joomla.org/How\\_do\\_you\\_put\\_a\\_module\\_inside\\_an\\_article%3F](https://docs.joomla.org/How_do_you_put_a_module_inside_an_article%3F). 27.4.2017.
- Joomshaper.com. 2015. Custom module above and below component. <https://www.joomshaper.com/documentation/helix-framework/helix3#custom-module-above-and-below-component>. 27.4.2017.
- Joomshaper.com. 2016a. Helix3 Framework. <https://www.joomshaper.com/helix>. 30.1.2017.
- Joomshaper.com. 2016b. A detailed guide to Helix3 Layout Manager. <https://www.joomshaper.com/blog/a-detailed-guide-to-the-helix3-layout-manager>. 15.5.2017.
- Kuusk, M. 2016. Verkkosivuston toteuttaminen sisällönhallintajärjestelmällä. Haaga-Helia Ammattikorkeakoulu. Tietojenkäsittelyn koulutusohjelma. Opinnäytetyö. <http://urn.fi/URN:NBN:fi:amk-201605056282>. 31.1.2017.
- Mediamyynti.karjalainen.fi. 2017a. <http://mediamyynti.karjalainen.fi/lyhyesti>. 20.5.2017.
- Mediamyynti.karjalainen.fi. 2017b. <http://mediamyynti.karjalainen.fi/ajankohtaista/tietoa>. 20.5.2017
- Oinonen, A. 2013. Verkkosivujen toteutus sisällönhallintajärjestelmällä. Mikkelin Ammattikorkeakoulu. Tietotekniikan koulutusohjelma. Opinnäytetyö. <http://urn.fi/URN:NBN:fi:amk-201302222522>. 2.2.2017.
- Repo, P. 2012. PK-yrityksen kotisivujen toteutus Joomla- julkaisujärjestelmällä. Pohjois-Karjalan Ammattikorkeakoulu. Tietotekniikan koulutusohjelma. Opinnäytetyö. <http://urn.fi/URN:NBN:fi:amk-201301281800>. 2.2.2017.
- Saajola, T. 2013. Joomla web-sivuston kehitysvälineenä. Turun Ammattikorkeakoulu. Tietojenkäsittelyn koulutusohjelma. Opinnäytetyö. <http://urn.fi/URN:NBN:fi:amk-2013060713494>. 31.1.2017.
- Schechter, E. 2016. Moving towards a more secure web. Google security blog. 8.9.2016. <https://security.googleblog.com/2016/09/moving-towards-more-secure-web.html>. 26.4.2017.
- Smith, D., Negrino, T. 2007. JavaScript Tehokas hallinta. Helsinki: Readme.fi.
- Toufiqur, R. 2017. How To Change Off-Canvas Position? Foorumi. 04.1.2017. <https://www.joomshaper.com/forums/how-to-change-off-canvas-position>. 11.1.2017.

- W3.org. 2011a. HTML/Elements/header.  
<https://www.w3.org/wiki/Html/Elements/header>. 15.5.2017.
- W3.org. 2011b. HTML/Elements/footer.  
<https://www.w3.org/wiki/Html/Elements/footer>. 15.5.2017.
- W3.org. 2014. HTML/Elements. <https://www.w3.org/wiki/HTML/Elements>.  
15.5.2017.
- Wiciak, M. 2014. Creating responsive ui for web store using CSS. Jyväskylän Ammattikorkeakoulu. Tietotekniikan koulutusohjelma. Opinnäytetyö.  
<http://urn.fi/URN:NBN:fi:amk-201405239584>. 6.2.2017.

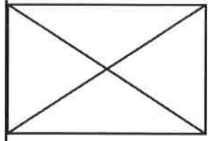
**Alkuperäinen idea templatesta**

	<b>Moduuli</b>	<b>Moduuli</b>	
	<b>Komponentti</b>		

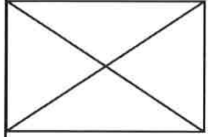
Wireframe kuvat etusivu, kategoria ja artikkeli



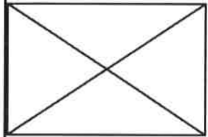




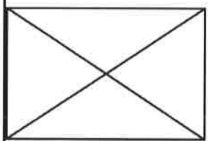
**Lorem ipsum dolor sit amet.**  
Consectetur adipiscing elit,  
sed do eiusmod tempor incididunt.



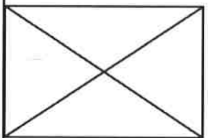
**Lorem ipsum dolor sit amet.**  
Consectetur adipiscing elit,  
sed do eiusmod tempor incididunt.



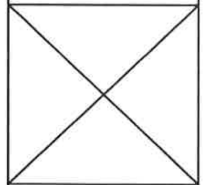
**Lorem ipsum dolor sit amet.**  
Consectetur adipiscing elit,  
sed do eiusmod tempor incididunt.

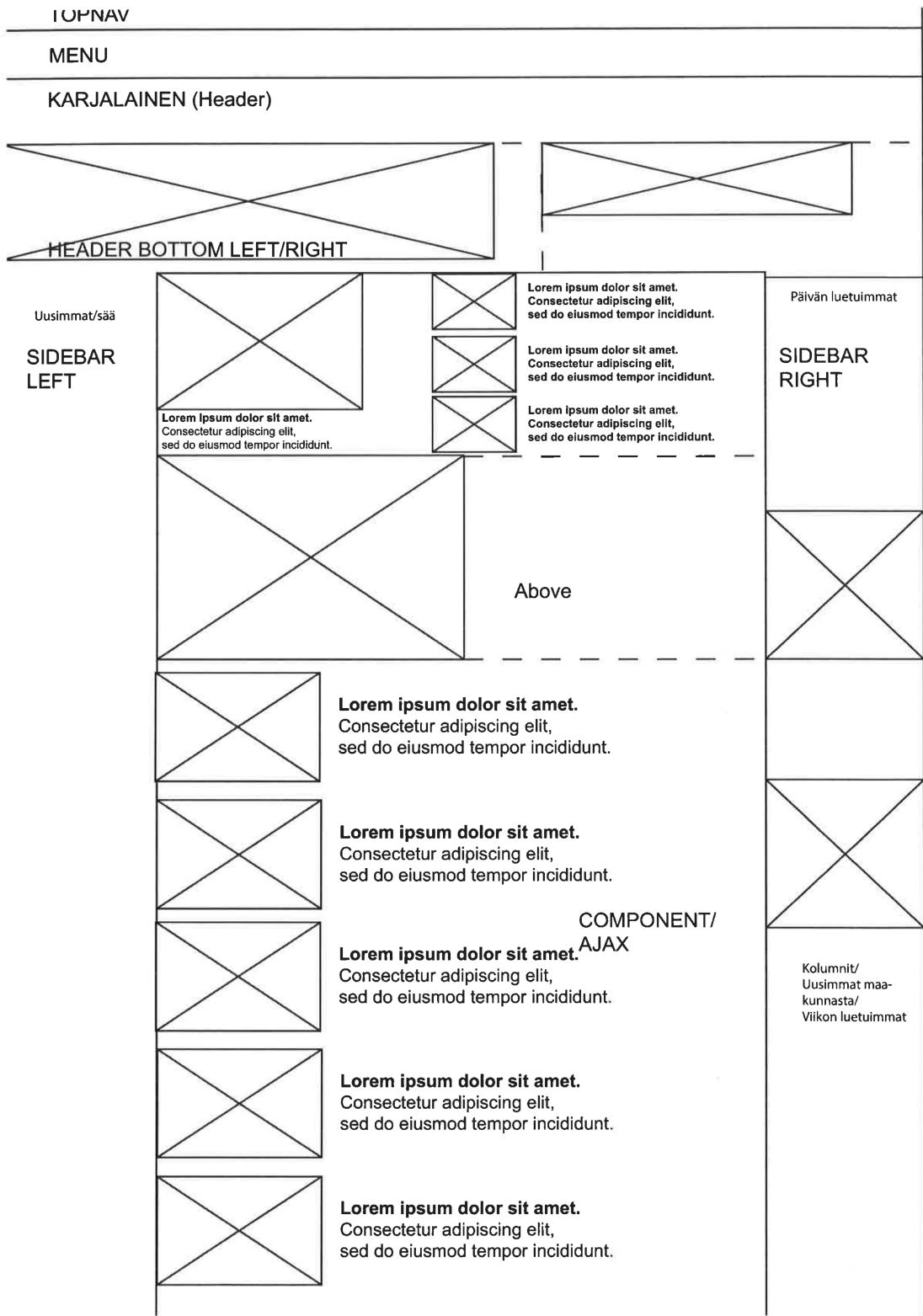


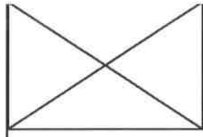
**Lorem ipsum dolor sit amet.**  
Consectetur adipiscing elit,  
sed do eiusmod tempor incididunt.



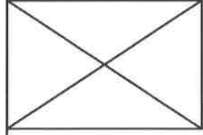
**Lorem ipsum dolor sit amet.**  
Consectetur adipiscing elit,  
sed do eiusmod tempor incididunt.



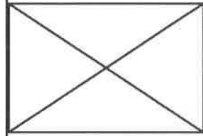




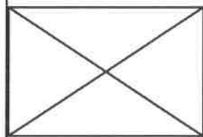
**Lorem ipsum dolor sit amet.**  
Consectetur adipiscing elit,  
sed do eiusmod tempor incididunt.



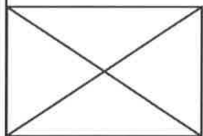
**Lorem ipsum dolor sit amet.**  
Consectetur adipiscing elit,  
sed do eiusmod tempor incididunt.



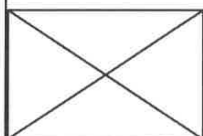
**Lorem ipsum dolor sit amet.**  
Consectetur adipiscing elit,  
sed do eiusmod tempor incididunt.



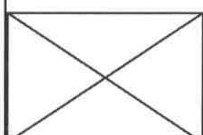
**Lorem ipsum dolor sit amet.**  
Consectetur adipiscing elit,  
sed do eiusmod tempor incididunt.



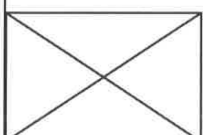
**Lorem ipsum dolor sit amet.**  
Consectetur adipiscing elit,  
sed do eiusmod tempor incididunt.



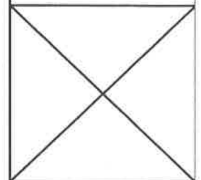
**Lorem ipsum dolor sit amet.**  
Consectetur adipiscing elit,  
sed do eiusmod tempor incididunt.



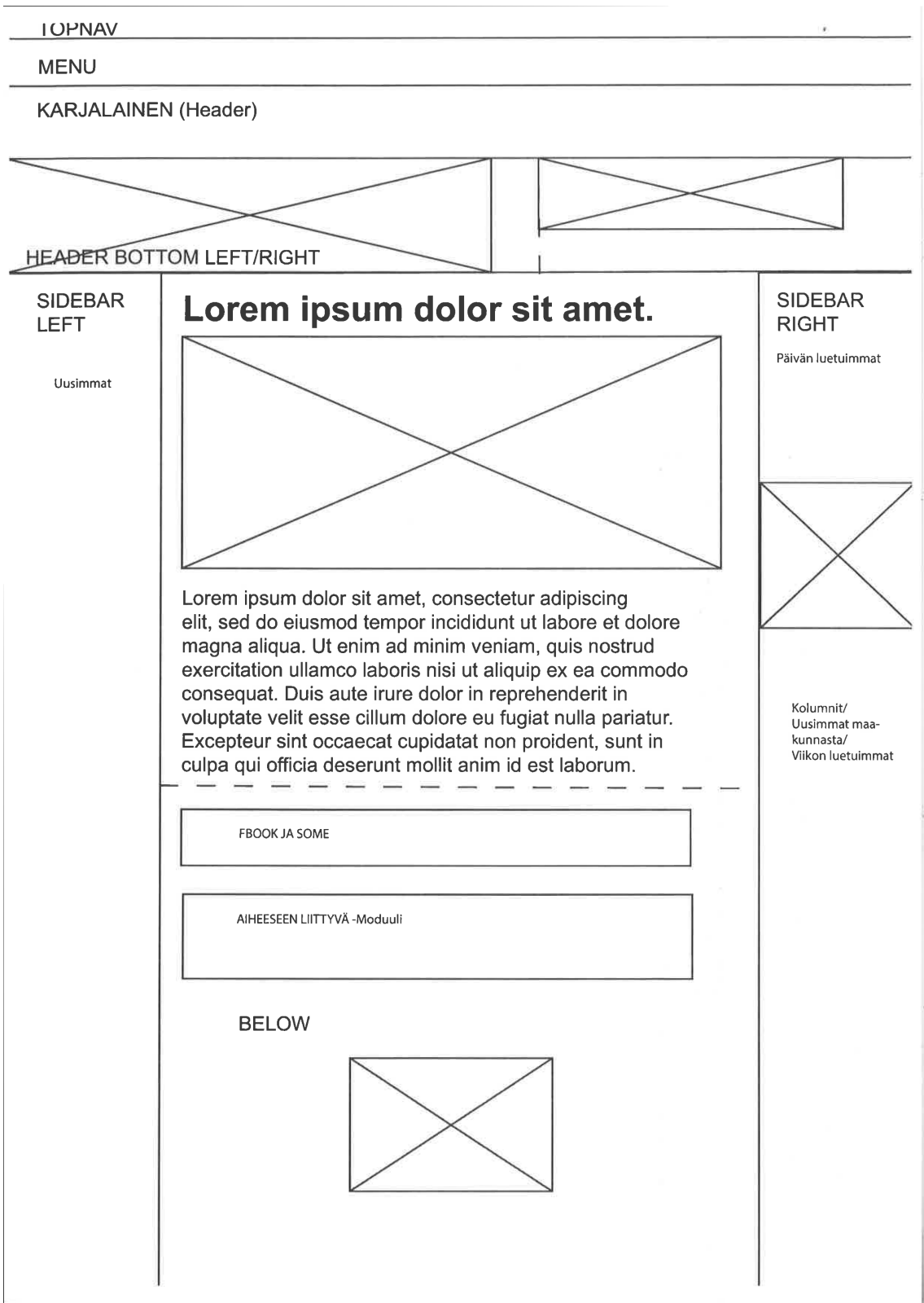
**Lorem ipsum dolor sit amet.**  
Consectetur adipiscing elit,  
sed do eiusmod tempor incididunt.

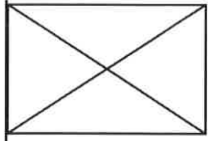


**Lorem ipsum dolor sit amet.**  
Consectetur adipiscing elit,  
sed do eiusmod tempor incididunt.

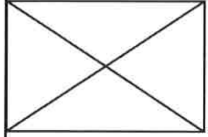


- 1
- 2
- 3
- 4
- 5

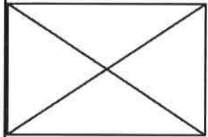




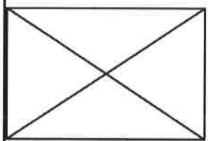
**Lorem ipsum dolor sit amet.**  
Consectetur adipiscing elit,  
sed do eiusmod tempor incididunt.



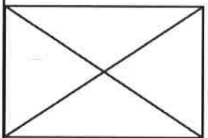
**Lorem ipsum dolor sit amet.**  
Consectetur adipiscing elit,  
sed do eiusmod tempor incididunt.



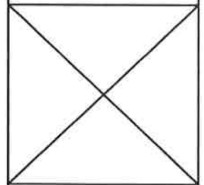
**Lorem ipsum dolor sit amet.**  
Consectetur adipiscing elit,  
sed do eiusmod tempor incididunt.



**Lorem ipsum dolor sit amet.**  
Consectetur adipiscing elit,  
sed do eiusmod tempor incididunt.



**Lorem ipsum dolor sit amet.**  
Consectetur adipiscing elit,  
sed do eiusmod tempor incididunt.



**Käsin piirretty malli adblock -moduulin toiminnasta**

