

Toni Nyström

# SilverStripen hyödyntäminen modernissa webkehityksessä

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Mediatekniikan koulutusohjelma

Insinöörityö

9.5.2017

Tekijä Otsikko	Toni Nyström SilverStripen hyödyntäminen modernissa webkehityksessä
Sivumäärä Aika	37 sivua 9.5.2017
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Mediatekniikka
Suuntautumisvaihtoehto	Digitaalinen media
Ohjaajat	Head of Digital Sampo Pihlaja Lehtori Olli Alm
<p>Insinööriyön tarkoituksena oli toteuttaa uusi verkkosivusto markkinointitoimistolle avoimen lähdekoodin sisällönhallintajärjestelmä SilverStripella. Insinööriyö lähti liikkeelle SilverStripen ja muutaman suosituimman avoimen lähdekoodin sisällönhallintajärjestelmän yleisellä vertailulla, minkä jälkeen kartoitettiin SilverStripen tarkempia ominaisuuksia ja työnkulkua. Verkkosivustoprojektiosion jälkeen toteutettiin verkkosivustovertilu ja analysoitiin uutta ja vanhaa sivustoa teknisestä näkökulmasta.</p> <p>Yleisessä vertailussa olivat mukana SilverStripen lisäksi WordPress, Joomla! ja Drupal, joiden vertailu tehtiin hyvin pintapuolisesti, eikä tarkempiin ominaisuuksiin kiinnitetty juuri-kaan huomiota. SilverStripen opetteluosiossa käytiin läpi SilverStripen asennus, teemojen ja sivupohjien luominen ja yleisiä käytäntöjä.</p> <p>Verkkosivustoprojekti toteutettiin yhteistyössä markkinointitoimiston muun henkilökunnan kanssa: graafinen suunnitelma tuli yrityksen graafiselta suunnittelijalta ja sivuston tekniset vaatimusmäärittelyt tehtiin koko yrityksen voimin. Sivustoprojektin tavoitteena oli luoda teknisesti ja visuaalisesti moderni ja responsiivinen sivusto, joka olisi kaikilta osin edistyneempi verrattuna vanhaan WordPressillä toteutettuun sivustoon. Sivuston rakennusprosessia käytiin läpi hyvin yksityiskohtaisesti, pääpainon ollessa SilverStripen ominaisuuksissa, mutta myös muihin moderneihin teknologioihin ja työtapoihin kiinnitettiin huomiota. Muita käytettyjä teknologioita olivat esimerkiksi Vagrant, Virtual Box, npm ja BitBucket.</p> <p>Verkkosivustovertilua varten toteutettiin kysely pääasiallisesti yrityksen sisällä. Kyselyn tuloksista selvisi, että uusi sivusto oli kaikilta osin parempi kuin vanha, erityisesti vaatimusmäärittelyiden mukaisilta osilta, joita olivat tekninen toimivuus ja visuaalisuus.</p> <p>Insinööriyön sovellusosio oli kaikilta osin onnistunut, ainoastaan aikataulusta oltiin jäljessä. Projektin tuloksena syntyi modernit standardit täyttävä näyttävä verkkosivusto, joka kuvastaa hyvin markkinointitoimiston ilmettä ja toimintatapoja. Sivuston kehitystä on tarkoitus jatkaa, ja tulevaisuudelle onkin jo mietittynä blogiosio ja tavoitteena on myös toteuttaa markkinoinnin automaatiota.</p>	
Avainsanat	SilverStripe, web, sisällönhallinta, webkehitys

Author Title	Toni Nyström Benefits of SilverStripe in Modern Web Development
Number of Pages Date	37 pages 9 May 2017
Degree	Bachelor of Engineering
Degree Programme	Media Technology
Specialisation option	Digital Media
Instructors	Sampo Pihlaja, Head of Digital Olli Alm, Senior Lecturer
<p>The main purpose of this thesis was to develop a new web site for the marketing agency using the open-source content management system SilverStripe. The thesis started off by a general comparison between SilverStripe and a few other popular open-source content management systems. After that a deeper research of the features and workflow of SilverStripe was done. When the web site was finished and published, the new and the old web site were compared and analyzed mainly from the technical perspective.</p> <p>The compared content management systems were – in addition to SilverStripe – WordPress, Joomla and Drupal. The comparison was made superficially and the explicit features were not included in the comparison. The research of SilverStripe included installation, creation of themes and templates and some general practices.</p> <p>The web site project was executed in collaboration with the staff of the marketing agency. The visual design was made by the company's graphic designer and the technical specifications were made by the whole company. The purpose of the web site project was to create a web site that is both technically and visually modern and responsive. In other words, a web site which would be more advanced in all levels compared to the old WordPress site. The building process of the web site was reviewed in a very detailed way. The main focus was in the features of SilverStripe but attention was also paid to other modern technologies and processes. Other technologies used were for example Vagrant, Virtual Box, nom and BitBucket.</p> <p>A survey was done mainly inside the company to compare the old and the new web site. The summary of the survey indicated that the new web site outdid the old in all areas, especially in the technical and visual operability.</p> <p>The web site part of the thesis was successful in all areas, although the project was published behind schedule. The final result of the project was a modern web site which meets all the modern standards. The development of the site is supposed to continue in the future with a blog and automated marketing.</p>	
Keywords	SilverStripe, web, content management, web development

## Sisällys

1	Johdanto	1
2	SilverStripe-sisällönhallintajärjestelmä	2
2.1	Sisällönhallintajärjestelmät yleisesti	2
2.2	Sisällönhallintajärjestelmien vertailu	3
2.3	SilverStripen ominaisuudet	4
2.4	Asennus ja toiminta	5
2.5	Teemat	6
2.6	Sivupohjat	7
2.7	Tyylitiedostot	8
2.8	Sisällönhallinta	8
3	Verkkosivut Inuitsille	11
3.1	Tehtävänanto	11
3.2	Yleisesti	12
3.3	Valmistelu	14
3.4	Toteutus	22
4	Projektin lopputulos ja sivustovertilu	32
5	Yhteenveto	36
	Lähteet	38

## Lyhenteet

CMS	<i>Content Managing System.</i> Sisällönhallintajärjestelmä.
CSS	<i>Cascading Style Sheets.</i> Verkkosivujen ulkoasun määrittelyyn käytetty tyylilohje.
HTML	<i>Hypertext Markup Language.</i> Verkkosivujen rakentamisessa yleisesti käytetty merkintäkieli.
MVC	<i>Model-View-Controller</i> -ohjelmistoarkkitehtuurimalli.
OO	<i>Object Oriented.</i> Olio-ohjelmointi.
ORM	<i>Object Relational Mapping.</i>
WYSIWYG	<i>What You See Is What You Get.</i> Käyttöliittymä, joka pyrkii esittämään lopputulosta mahdollisimman tarkkaan vastaavan näkymän jo editointivaiheessa.

## 1 Johdanto

Insinööriyön tavoitteena on luoda projektina uudet kotisivut yritykselle Interactive Inuits Oy, markkinointinimeltään Inuits, jolle työskentelen vakituisesti. Kotisivut rakennetaan SilverStripe-sisällönhallintajärjestelmän pohjalta, ja projekti toimii omatoimisena perehdytyksenä Inuitsin käyttämiin teknologioihin. Kotisivujen päivittäminen uuteen versioon on tärkeää Inuitsin kaltaiselle markkinointitoimistolle, jonka pääpaino on digitaalisen sisällön ympärillä. Sivuston päivitys on osa suurempaa brändiuudistusta, jossa yrityksen käyttämä nimi Inuit vaihtuu uuteen nimeen Inuits.

Sivustoprojektin alustamista varten tarkoituksena on tutkia SilverStripen vahvuuksia ja heikkouksia vertailemalla sitä kolmeen muuhun yleisesti käytettyyn sisällönhallintajärjestelmään. Vertailun jälkeen tarkoituksena on tutustua tarkemmin SilverStripen ominaisuuksiin ja kerätyn tiedon pohjalta rakentaa verkkosivusto SilverStripeä hyödyntäen. Lisäksi tarkoituksena on analysoida ja vertailla uutta ja vanhaa sivustoa kyselyn avulla.

Esiselvityksen tavoitteena on vertailla neljää avoimen lähdekoodin sisällönhallintajärjestelmää, WordPressiä, Joomlaa, Drupalia ja SilverStripeä, ja esitellä niiden ominaisuuksia. Esiselvityksen toisessa osassa keskitytään SilverStripeen tarkemmin, perehdytään sen tarkempiin ominaisuuksiin sekä tutustutaan sen käyttöön ja työnkulkuun.

Kotisivujen julkaisun jälkeen tarkoituksena on luoda kysely ja tehdä kerättyjen käyttökemuksien perusteella arvio uuden sivuston ja alustan toimivuudesta verrattuna aiempaan WordPressillä toteutettuun sivustoon. Kysely toteutetaan yrityksen sisällä, ja tavoitteena on kerätä arvioita asiantuntijanäkökulmasta niin ohjelmointi- kuin markkinointinäkökulmasta.

Insinööriyön on tarkoitus toimia omatoimisena perehdytyksenä SilverStripeen, minkä jälkeen siirryn hyödyntämään työssä oppimiani taitoja Inuitsin oikeissa asiakasprojekteissa.

## 2 SilverStripe-sisällönhallintajärjestelmä

### 2.1 Sisällönhallintajärjestelmät yleisesti

Sisällönhallintajärjestelmä on pääasiallisesti verkossa toimiva järjestelmä, jolla pyritään helpottamaan verkkosisällön, eli tekstien ja median, luomista ja ylläpitoa. Käsitteenä sisällönhallintajärjestelmä on hyvin laaja ja voi pitää sisällään minkä tahansa digitaalisen sisällön hallinnointia esimerkiksi suurissa yrityksissä. Tässä insinööriyössä keskitytään, ja sisällönhallintajärjestelmällä viitataan, verkkosisällön hallintajärjestelmään. (1.)

Monissa sisällönhallintajärjestelmissä on myös työkaluja webkehittäjän työn, eli verkkosivuston rakentamisen, nopeuttamiseksi ja helpottamiseksi. Muun muassa WordPressissä verkkosivuston rakentaminen on tehty valmiiden pohjien ja lisäosien avulla niin helpoksi, ettei käyttäjä tarvitse juuri lainkaan ohjelmointikokemusta.

Sisällönhallintajärjestelmät voidaan lähdekoodinsa perusteella jakaa pääasiallisesti kahteen eri luokkaan: avoimen lähdekoodin järjestelmiin (*open source*) ja suljetun lähdekoodin järjestelmiin (*closed source*).

Avoimen lähdekoodin järjestelmissä järjestelmän taustaohjelmistojen lähdekoodit ovat julkisia ja jokainen voi tarvittaessa tehdä niihin omia paikallisia muutoksia. Avoimen lähdekoodin järjestelmissä käyttäjät ja ulkopuoliset kehittäjät voivat olla mukana järjestelmän kehittämisessä. Suurin motivaatio avoimen lähdekoodin järjestelmien käyttöön lieenee kuitenkin se, että järjestelmät ovat yleensä maksutta käytettävissä. Tästä syystä nämä sisällönhallintajärjestelmät ovatkin erittäin suosittuja pienempien sivustoprojektien yhteydessä. Avoimeen lähdekoodiin perustuvia sisällönhallintajärjestelmiä ovat esimerkiksi tässä työssä vertailussa olevat WordPress, Joomla!, Drupal ja SilverStripe.

Suljetun lähdekoodin järjestelmien lähdekoodit ovat piilotettuja, eikä ulkopuolisilla kehittäjillä ole pääsyä niihin. Suljetun lähdekoodin järjestelmissä käyttäjällä on usein tiiviimpi yhteistyö järjestelmän kehittäjän kanssa ja järjestelmä on myös tietoturvasempi. Näiden järjestelmien käytöstä peritään usein korkeita lisenssimaksuja, minkä takia ne eivät usein sovellu pienemmille yrityksille. Suljetun lähdekoodin sisällönhallintajärjestelmiä ovat esimerkiksi Kotisivukone ja SharePoint.

## 2.2 Sisällönhallintajärjestelmien vertailu

Sisällönhallintajärjestelmät on aina rakennettu johonkin tiettyyn tarkoitukseen, esimerkiksi blogi-, verkkosivu- tai suuremmaksi järjestelmäalustaksi. Vertailuryhmästä ei voida nostaa esiin yhtä järjestelmää ylitse muiden, vaan voidaan ainoastaan etsiä haluttuun käyttötarkoitukseen sopivinta sisällönhallintajärjestelmää.

WordPress on alun perin tarkoitettu lähinnä blogialustana toimivaksi, mutta sen käyttö on laajentunut myös tavallisiin verkkosivustoihin ja osaksi suurempia järjestelmäkokoaisuuksia. WordPressin suurimpana valttina ovat tuhannet maksulliset ja maksuttomat ulkoasuteemat ja lisäosat, joiden avulla verkkosivusto voidaan saada kokoon ilman minikäänlaista ohjelmointikokemusta. Useimmat teemat ovat suoraan mobiili- ja hakukoneoptimoituja. Merkittävänä etuna voidaan myös pitää sivuston hallintapuolta, joka on onnistuttu pitämään päällisin puolin yksinkertaisena. WordPress tarjoaa myös maksuttoman URL-osoitteen, joka ovat muotoa `http://sivustonnimi.wordpress.com`. Koska WordPress on alun perin blogialusta, on sen lähdekoodi ja ohjelmistoarkkitehtuuri edelleen hyvin alkeellinen. Modernin webkehityksen kannalta WordPressin kompastuskiviä ovat OO-ohjelmoinnin ja MVC-arkkitehtuurin puuttuminen suuresta osasta WordPressin rakenteita (2). Päivitysten yhteydessä liitännäisten toiminta järjestelmän kanssa saattaa häiriintyä ja koko sivusto mennä rikki. Omien liitännäisten tekeminen ja ylläpitäminen on haastavaa WordPressin taustajärjestelmän puutteiden vuoksi. Kaikesta huolimatta WordPress on maailman suosituin avoimen lähdekoodin sisällönhallintajärjestelmä yhteensä noin 140 miljoonalla latauksella (3).

Joomla on toiseksi suosituin heti WordPressin jälkeen 63 miljoonalla latauksella (3). Joomla on vaikeampi käyttää kuin WordPress, mutta silti edelleen varsin yksinkertainen. Sivustojen muokkaus onnistuu kätevästi useilla tuhansilla erilaisilla teemoilla, eikä ohjelmointikokemusta tarvitse välttämättä lainkaan. Joomlaan pääpaino on sosiaalisissa ja verkkokauppaomaisissa sivustoissa, johon se toimii erittäin hyvin. Joomla on teknisesti vähintään samalla tasolla kuin WordPress, muttei kuitenkaan saavuttanut yhtä laajaa kannattaja- ja käyttäjäkuntaa kuin WordPressin.

Drupal on kolmanneksi suosituin avoimen lähdekoodin sisällönhallintajärjestelmä 15 miljoonalla latauksella, ja se on teknisesti hyvin edistynyt, mutta samalla myös haastavampi oppia (3). WordPressiin ja Joomlaan verrattuna Drupal on erittäin tekninen, mutta sa-



malla myös joustavampi ohjelmointialusta, jonka käyttöä varten tarvitsee hallita vähintään yleisten webohjelmointikielten perusteet. Drupalin käytön vaikeuden takia voi olla vaikea löytää osaavia ohjelmoijia järkihintaan, ja monesti varsinkin yksinkertaisen WordPress-sivuston voi saada huomattavasti halvemmalla. Suurissa järjestelmissä Drupal tulee kuitenkin halvemmaksi. Drupalissa on myös saatavilla valmiita teemoja ja liitännäisiä. (4).

### 2.3 SilverStripen ominaisuudet

SilverStripe on ilmainen, avoimen lähdekoodin PHP-webkehityksen ohjelmistokehys ja sisällönhallintajärjestelmä. Se on pääasiassa uusiseelantilaisen SilverStripe Ltd.:n kehittämä ja ylläpitämä ja myös ensimmäinen avoimen lähdekoodin websovellus, jolle on myönnetty Microsoft-sertifikaatti. (5; 7.) Se on myös voittanut 20 IT-alan palkintoa (6). SilverStripe on julkaistu avoimen lähdekoodin BSD-lisenssillä (*Berkley Software Distribution*), mikä tarkoittaa, ettei sen käyttöä ole juurikaan rajoitettu, vaan sitä saa käyttää ja muokata hyvin vapaasti.

SilverStripe CMS on rakennettu SilverStripe-ohjelmistokehityksen (SilverStripe Framework) päälle. SilverStripe Framework on olioperustainen (*object-oriented*), PHP5:een pohjautuva MVC/ORM-ohjelmistokehys. (7.) Tässä työssä termillä SilverStripe viitataan SilverStripeen kokonaisuutena: se sisältää siis SilverStripe CMS:n ja SilverStripe Frameworkin.

SilverStripe, kuten monet muutkin sisällönhallintajärjestelmät, perustuu laajasti käytettyihin koodikieliin (HTML, CSS, PHP, JavaScript), ja se vaatii tietokannan toimiakseen. SilverStripe on alustana erittäin joustava ja sopiikin monille erityyppisille sivustoprojekteille käyntikorttisivustoista verkkokauppoihin. SilverStripe-sivustoja ei ole pakotettu noudattamaan tiettyjä ulkoasullisia kaavoja tai teemoja. Esimerkiksi WordPressissä ja Joomla:ssa ulkonäköä ja toiminnallisuuksia ohjataan pääasiassa erillisillä teemoilla ja lisäosilla, joiden muokkaaminen on usein aikaa vievää. Näissä järjestelmissä esimerkiksi oman JavaScript-koodin lisäys on monin paikoin huomattavasti monimutkaisempaa kuin SilverStripessä, jossa koodin voi kirjoittaa suoraan HTML:n sekaan ilman, että järjestelmän toiminta häiriintyy. SilverStripe toimiikin yleensä helposti erilaisten JavaScript-ohjelmistokehysten ja -kirjastojen kanssa, kuten esimerkiksi AngularJS:n. Pian julkaistavaan SilverStripe 4:ään on myös sisällytetty ReactJS. (8; 9.)

## 2.4 Asennus ja toiminta

SilverStripe vaatii toimiakseen palvelimen, joka voi olla julkinen tai paikallinen (localhost). Eri alustavaihtoehtoja on paljon, mutta Linuxin Apache on suositeltu. (10.) Asiakasprojektissa SilverStripe asennettiin Vagrantin virtuaaliympäristöön. Palvelimelle on asetettu tietyt vaatimukset, jotka ovat viimeisimmässä SilverStripe 3.4:ssä PHP 5.3.3+, MySQL 5.0+ ja Apache 2.0+. Myös muita vaihtoehtoja on käytettävissä.

SilverStripen asennus onnistuu perinteisesti lataamalla tiedostot SilverStripen asennussivulta ja siirtämällä tiedostot palvelimelle. Suositeltu asennustapa on kuitenkin asennus Composerin avulla. Composer on työkalu, jolla voidaan hallita PHP-riippuvuussuhteita (*dependency management*). Se on pakettienhallintatyökalu, jonka avulla voidaan asentaa haluttuja ohjelmakokonaisuuksia ja -kirjastoja palvelimelle. Samaan tapaan toimivat myös Noden npm ja Rubyn bundler. Composerin avulla kaikki asennetut kokonaisuudet ja kirjastot pysyvät järjestyksessä ja niitä on helppo päivittää. (13.)

Tiedostojen asennuksen jälkeen vaaditaan tietokantojen konfigurointi, joka onnistuu muokkaamalla projektin juurihakemistosta löytyvää `_ss_environment.php`-tiedostoa, johon tulee syöttää tarvittavat tietokantatiedot ja CMS:n hallintapuolen käyttäjätunnus ja salasana. Kun `_ss_environment.php` on oikein konfiguroitu, voidaan SilverStripe asentaa tietokantaan.

SilverStripe-projektissa taustakoneisto, eli *back-end*, on eriytetty sivuston graafisesta puolesta, eli front-endistä. Tämä on esitetty taulukossa 1. Vastaavantyyppinen jaottelu on yleistä myös muissa sisällönhallintajärjestelmissä.

Taulukko 1. SilverStripen hakemistorakenne (11).

Hakemisto	Kuvaus
<code>assets/</code>	Sisältää CMS:n kautta ladatut kuvat ja muut tiedostot.
<code>cms/</code>	Sisältää kaikki CMS:n eli sivuston hallintapuolen tiedostot.
<code>framework/</code>	Ohelmistokehys, joka rakentaa sekä sivuston, että hallintapuolen.

<i>mysite/</i>	Sisältää sivuston back-end-koodit, pääasiassa PHP:n.
<i>themes/</i>	Yhdistää kaikki kuvat, tyyli tiedostot, JavaScript-koodit ja sivumallit toimivaksi teemaksi.

Sivustoa rakennettaessa muokataan pääasiassa ainoastaan *assets-*, *mysite-* ja *themes-* hakemistojen sisältöä. *Mysite-* hakemiston tiedostot ovat ainoastaan back-endiä ja kaksi muuta front-endiä. *Cms-* ja *framework-* hakemistot ylikirjoittuvat SilverStripea päivitettäessä, minkä vuoksi niiden sisältöjen muokkaamista tulee välttää.

## 2.5 Teemat

SilverStripen teemat toimivat samalla periaatteella kuin muissa sisällönhallintajärjestelmissä. Teemalla tarkoitetaan kokonaisuutta, joka koostuu erilaisista sivupohjista ja niihin liittyvistä tyyliohjeista ja muista tarvittavista koodista. Yhdellä sivustolla voi olla useita eri teemoja, kuten pöytäkone- (*desktop*) ja mobiiliteema. WordPressin ja Joomla:n tavoin myös SilverStripelle on olemassa valmiita teemoja, joita voi käyttää omissa projekteissa. Tässä työssä keskitytään kuitenkin sivuston rakentamiseen alusta lähtien.

Uuden teeman luomiseksi luodaan uusi hakemisto *themes/*-hakemiston alle. Hakemisto nimetään uuden teeman nimen mukaiseksi, joka tässä asiakasprojektissa on "inuit". Hakemistoon voidaan luoda uusia hakemistoja, esimerkiksi kuville ja JavaScript-tiedostoille. Merkittävintä on kuitenkin luoda uusi sivupohjahakemisto *templates/*, jonka alle luodaan hakemistot *Includes/* ja *Layout/*. Hakemiston *templates/* alle luodaan tiedosto *Page.ss*, joka toimii sivuston perussivupohjana.

Tiedostopääte *.ss* viittaa SilverStripen omaan tiedostomuotoon, joka on yksinkertaisesti pelkkä tekstitiedosto, joka suoritetaan SSVIEWER-jäsentäjän (*parser*) läpi, jolloin lopputuloksena on koodia. SS-tiedosto voi sisältää yleisimpiä koodikieliä, esimerkiksi HTML, XML, JSON ja JavaScript. Yleensä SS-tiedosto sisältää kuitenkin ainoastaan HTML-koodia. SS-tiedostot toimivat SilverStripen sivupohjina (*template*), ja käyttäjälle selaimessa näkyvä osa koostuu näistä sivupohjista. Mahdollisuus linkittää sivupohjia toisiinsa yksinkertaisella *Include*-koodilla tekee SilverStripestä modulaarisen ja vähentää täten turhaa työtä.

Sivutyypin toimivuus vaatii kuitenkin projektin teeman määrittämisen, eli SSVIEWER-jäsentäjä tarvitsee tiedon siitä, minkä teeman sivupohjia sen kuuluu kääntää. Projektilla on mahdollista olla useita teemoja, ja niitä voi aktivoida esimerkiksi kampanjatyylisesti. SSVIEWER on polussa `mysite/_config/config.yml` otsikolla "SSVIEWER". (12.) Teeman määrittäminen tehdään esimerkikoodin 1 mukaisesti.

```
***config.yml
...
SSVIEWER:
  theme: 'inuit'
```

Esimerkkikoodi 1. Teeman määrittäminen SSVIEWER-jäsentäjälle.

## 2.6 Sivupohjat

Teeman luomisen jälkeen on mahdollista luoda lisää sivupohjia. Perinteisellä verkkosivustolla on muutamia eri sivupohjia. Näitä ovat esimerkiksi asiakastyössä esiintyvät etu- ja projektisivut, joissa molemmilla sivuilla on omat, toisistaan poikkeavat sisältöalueet. Etusivulla on mahdollisesti lyhyt yleisesittely ja uusimpien projektien listaus, jossa jokaisesta projektista näytetään ainoastaan otsikko ja pääkuva. Projektisivulla sen sijaan on paljon tekstiä, mutta jokaisessa projektissa sisältö esitetään samassa muodossa. SilverStripessa molemmille sivupohjille täytyy luoda omat sivutyypit polun `themes/inuit/Layout` hakemistoon. Luodaan uudet tiedostot: etusivua varten `FrontPage.ss` ja projektisivua varten `ProjectItemPage.ss`. Sivutyypit voidaan nimetä halutulla tavalla, mutta tiedostonimen tulee alkaa isolla alkukirjaimella ja käytössä ovat ainoastaan kirjaimet välillä A–Z ja a–z. `FrontPage.ss`- ja `ProjectItemPage.ss`-tiedostoihin voidaan tämän jälkeen syöttää halutut HTML-koodit.

Esimerkin etu- ja projektisivut ovat sisältöalueiltaan erilaiset, mutta niissä on monesti myös samoja ominaisuuksia. Useiden verkkosivustojen jokaisella sivulla on yhteinen logo ja valikon sisältävä yläpalkki ja yhteystiedot tai sivukartat sisältävä alapalkki. Palkit voidaan tehdä erikseen joka sivulle, mutta varsinkin laajemmissa sivukokonaisuuksissa tämä on aikaa vievää ja pienet tekstimuokkaukset joudutaan tekemään jokaiselle sivulle erikseen. Tätä helpottaa `templates/-`hakemiston `Page.ss`, joka on muiden sivujen pohjana toimiva pääsivupohja. Pääsivuun voidaan syöttää kaikki elementit, jotka toistuvat

jokaisella sivulla. Muilla sivutyypeillä määritetään tarkemmin ainoastaan sivun sisältöalueen rakennetta. Page.ss-tiedostoon voidaan määritellä muiden sivupohjien sisällön paikka muuttujalla `Layout`.

## 2.7 Tyylitiedostot

Tyylitiedostojen käyttäminen SilverStripessa toimii samalla tavalla kuin tavallisellakin HTML-sivustolla. On mahdollista luoda style.css-tiedosto suoraan themes/inuit/-hakemistoon ja määritellä jokaisen sivun tyylit samassa tiedostossa. Tämä johtaa yleensä hyvin pitkään ja vaikealukuiseen CSS-tiedostoon, minkä vuoksi tyylitiedostot on järkevämpää pilkkoa pienempiin osiin sivupohjien mukaisesti. Tyylitiedostoja varten voidaan luoda uusi hakemisto themes/inuit/-hakemiston alle, esimerkiksi nimellä `css/`. Tavallisen CSS:n käyttö on kuitenkin laajemmissa projekteissa hidasta, ja koodista tulee väkisinkin vaikealukuista ja vaikeammin päivitettävää. Asiakasprojektissa tulen käyttämään SASS:a, joka on CSS:n pohjalta kehitetty tyyliohje, jolla esimerkiksi väreille voidaan määrittää muuttujia ja porrastaa tyyliä. Aikaisemmissa projekteissa olen huomannut SASS:n olevan korvaamaton apu ja hyödyllisempi kuin CSS. Yleisen selvyuden vuoksi esittelen tässä insinööriyössä tarvittavat tyyliohjeet CSS-koodina SASS-koodin sijaan.

## 2.8 Sisällönhallinta

Esimerkeissä läpikäytyt sivupohjat ovat staattisia HTML-sivuja, joiden sisältöpäivitykset onnistuvat ainoastaan suoraan koodia muokkaamalla. Jotta sivupohjat saadaan liitettyä sisällönhallintajärjestelmään, täytyy sitä varten kertoa järjestelmän back-endille, mitkä ovat sivupohjia ja miten niiden kuuluu käyttäytyä. Tätä varten luodaan säätimiä (*controller*).

Controller-luokka on vastuussa oikean SS\_HTTP-vastauksen (SS\_HTTPResponse) toimittamisesta tiettyyn SS\_HTTP-pyyntöön (SS\_HTTPRequest) perustuen. SilverStripen SS\_HTTPRequest ja SS\_HTTPResponse toimivat pitkälti samalla tavalla kuin PHP:n HTTP-pyyntö ja -vastaukset: HTTP-vastaus toimittaa HTTP-pyyntöön liittyvät sisällöt, tavallisimmin siis verkkosivun. Controller-luokalla voidaan siis hallinnoida, mitä käyttäjälle näytetään sivustolla vierailtaessa. Se kertoo esimerkiksi, mitä toimintoja sivulla on ja mitkä sivupohjat tulee hahmontaa (*render*).

Sivutyypin luominen back-endiin onnistuu polussa `mysite/code`, johon luodaan uudet tiedostot samoilla nimillä, kuin niitä vastaavat sivupohjat on nimetty. Uudet tiedostot ovat PHP-tiedostoja, ja ne ovat esimerkin mukaan muotoa `Page.php`, `FrontPage.php` ja `ProjectItem.php`, jossa `Page.php` on edelleen perussivutyyppi ja loput täydentävät sitä. Kaikki sivutyypit noudattavat samaa kaavaa, ja ne koostuvat yleisesti kahdesta osiosta: tietokantakenttien ja -muuttujien määrittely muuttujassa `$db` ja tietokannan liittäminen hallintapaneeliin funktiolla `getCMSFields`. Tietokantamuuttuja on muodoltaan *arraylist*, joka voi pitää sisällään erityyppisiä alkioita. Käytettävissä olevat datatypit on esitelty taulukossa 2.

Taulukko 2. Tietokantamuuttujassa käytössä olevat datatypit.

<b>Boolean</b>	Totuusarvomuuuttujakenttä, 0 tai 1.
<b>Currency</b>	Numerokenttä kahden desimaalin tarkkuudella. Suunniteltu varastoimaan valuutta-arvoja.
<b>Date</b>	Päiväyskenttä, muoto DD/MM/YYYY tai YYYY-MM-DD.
<b>Decimal</b>	Desimaalitenttä.
<b>Enum</b>	String-listakenttä.
<b>HTMLText</b>	HTML-tekstikenttä, koko enintään 2 MB. Sisällyttää HTML-tagit.
<b>HTMLVarchar</b>	HTML-tekstikenttä, pituus enintään 255 merkkiä. Sisällyttää HTML-tagit
<b>Int</b>	Kokonaislukukenttä, 32 bittiä.
<b>Percentage</b>	Prosenttilukukenttä, määrittelyväli 0–1.
<b>SS_Datetime</b>	Päiväys- ja aikakenttä, muoto DD/MM/YYYY tai Y-m-d H:i:s
<b>Text</b>	Tekstikenttä, enimmäiskoko 2 MB. Suunniteltu varastoimaan raakatekstiä.
<b>Time</b>	Aikakenttä, muoto H:i:s.
<b>Varchar</b>	Tekstikenttä, maksimipituus 255 merkkiä. Suunniteltu varastoimaan raakatekstiä.

Tietokantakenttien funktiolla `getCMSFields` hallintapaneeliin liittämistä varten on olemassa monta erilaista lomakekenttaluokkaa (*FormField types*). Funktion `getCMSFields` päätoiminnallisuuksiin kuuluvat hallintapaneelikenttien hakeminen, niiden muokkaaminen ja palauttaminen. Funktiossa voi myös vaikuttaa esimerkiksi hallintapaneelikenttien järjestykseen ja kuvausteksteihin ja kenttiä voi jakaa usealle välilehdelle.

Taulukko 3. Yleisimpiä lomakekenttaluokkia tietokantakenttien liittämiseksi SilverStripen hallintapaneeliin.

<b>CheckboxField</b>	Yksittäinen valintaruutukenttä.
<b>TextField</b>	Yksirivinen tekstikenttä.
<b>TextareaField</b>	Monirivinen tekstikenttä.
<b>PasswordField</b>	Input-kenttä, teksti naamioitu.
<b>DateField</b>	Päivämääräkenttä. Mahdollista esittää erillisinä päivä-, kuukausi- ja vuosikenttinä tai kalenteriponnahdusikkunana.
<b>EmailField</b>	Sähköpostikenttä, jossa automaattinen sähköpostiformaatin validointi.
<b>HtmlEditorField</b>	WYSIWYG-editorikäyttöliittymä.
<b>NumericField</b>	Tekstikenttä, jossa validointi numeroita varten.
<b>PhoneNumberField</b>	Puhelinnumerokenttä, josta on mahdollista erotella puhelin- ja suuntanumero.
<b>GridField</b>	Esittää kenttiä taulukkomuodossa. Tarkoitettu esimerkiksi Data-Object-objektien esittämiseen.

### 3 Verkkosivut Inuitsille

#### 3.1 Tehtävänanto

Uusien kotisivujen tarve on Inuitsissa ajankohtainen, sillä tavoitteena on tehdä laajempi brändiuudistus yrityksen sisällä. Brändiuudistukseen kuuluu yrityksen markkinointinimen muuttaminen nimestä "Inuit" nimeen "Inuits", logon ja yleisilmeen muuttaminen sekä kotisivujen ja muun markkinointimateriaalin ilmeen uudistaminen.

Nykyinen kotisivu on ollut käytössä vasta vuoden, mutta digimarkkinointialalla sitä voidaan pitää jo vanhentuneena. Nykyinen sivusto on toteutettu WordPressillä ja teknisesti se on hyvin nykyaikainen, minkä vuoksi uusimistarve ei ole niinkään vanhentuneen tekniikan korvaamista vaan ilmeen uudistamista. Alustana WordPress toimii hyvin nykyisellä sivustolla, mutta Inuitsin muissa projekteissa on päätetty ottaa käyttöön SilverStripe, jonka takia Inuitsin uudesta sivustosta haluttiin tehdä ikään kuin esimerkkisivu asiakkaita varten.

Uusi sivusto toteutetaan SilverStripellä, ja graafiseen toteutukseen kiinnitetään erityistä huomiota. Tarkoituksena on luoda sivusto, joka kuvaa paremmin Inuitsin rentoa ja ammattitaitoista asennetta. Uusi sivusto tulee myös olemaan sisällöllisesti tyypistetympi ja yksinkertaistetumpi versio nykyisestä. Yrityksen tekemistä projekteista nostetaan esiin vain murto-osa, ja tarkoituksena onkin keskittyä esittelemään ainoastaan tärkeimpiä ja merkittävimpiä asiakkaita ja näyttävimpiä töitä.

Sivustoprojekti toteutetaan lähes koko yrityksen henkilöstön voimin, ja eri osa-alueet on jaettu eri henkilöiden toteutettavaksi. Tämä on hyvä menettelytapa, mutta riskinä on projektin pidentyminen aikataulullisesti liian monen liikkuvan osan takia. Sivuston ulkoasun toteutus tulee Inuitsin graafikolta Arthur Kavanderilta, joka suunnittelee myös uuden logon ja kuvakkeet, ja päättää, mitä kuvia sivustolle laitetaan. Lopulliset tekstit tulevat pääasiassa copywriter Aurinna Weismanilta projektipäälliköiden avustuksella. Sivuston uusimisen projektipäällikkönä toimii taiteellinen johtaja Markus Tamminen, joka toteuttaa myös sivustolla käytettävät showreel- ja projektivideot. Omiin tehtäviini kuuluu sivuston tekninen toteutus, lukuun ottamatta projektin teknistä pohjaa, jonka Lead Developer Teemu Lahti luo valmiiksi Bitbucketiin projektin aloitusta helpottamaan. Hänen luomansa pohja sisältää esiasennetun SilverStripen virtuaalikoneen päälle, SASS-esiprosessorin



ja muita työskentelyä nopeuttavia työkaluja. Muilta osin teknisen toteutuksen suhteen minulla on täydet vapaudet. Projektille ei ole asetettu rajoitteita tai tiettyjä raameja, mutta lopputuloksen tulee vastata sille annettua graafista suunnitelmaa.

### 3.2 Yleisesti

Ennen teknisen osion toteutuksen aloittamista kävimme graafikon tekemää graafista suunnitelmaa läpi useamman kerran ja teimme siihen muutoksia. Lopulta saimme hiottua alustavan graafisen suunnitelman huippuunsa ja Arthur Kavander toimitti minulle siitä Photoshop-version, jossa kaikki sivuston sivut ja osiot olivat omina kansioinaan ja tasoinaan.

Sivusto koostuu kolmesta erilaisesta sivupohjasta: etusivu, projektien yleisesittelysivu ja yksittäisen projektin tarkempi esittelysivu. Etusivu on pitkä sivu, jossa on monta eri osiota: esittelyvideo ja -teksti, projektien yleisesittely, palveluesittelyosio, henkilöstöesittely ja yhteystiedot sekä Google-kartta. Projektien esittelysivu kerää kaikki esitellyt projektit samalle sivulle. Projektin esittelysivulla projekti esitellään tarkemmin esittelytekstillä, videoilla ja kuvilla. Kaikille sivuille yhteistä on sivun yläosan banneri, joka sisältää joko täysleveän 16:9-taustavideon (kuva 1) tai -kuvan (kuva 2), Inuitsin logon ja navigaation avausnapin. Navigaatio päätettiin esteettisistä syistä toteuttaa hieman poikkeavalla tavalla, jossa yleensä mobiilissa käytetty hampurilaiskuvake on käytössä myös työpöytäversiossa.



Kuva 1. Esimerkki Inuits.fi:n etusivun banneriosiesta. Taustalla HTML5-video.



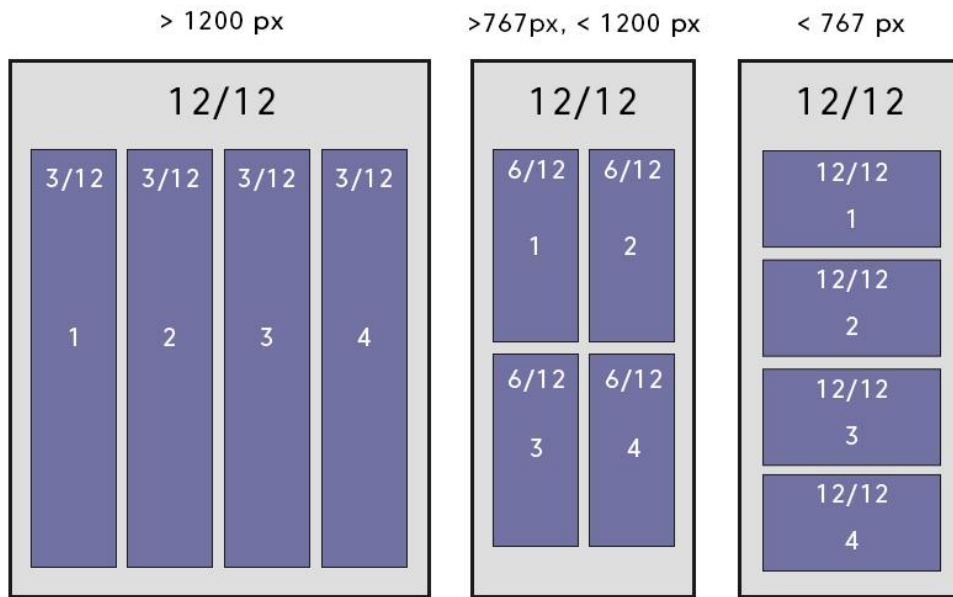
Kuva 2. Esimerkki Inuits.fi:n projektisivun banneriosiesta. Taustalla staattinen taustakuva.

### 3.3 Valmistelu

Projekti alkoi osaltani graafisen suunnitelman tarkalla läpikäynnillä ja teknisten tarpeiden tarkalla määrittelyllä. Selvitin muun muassa, mitä osioita sivustolta löytyy, mitä osioita voidaan kierrättää ja minkätyyppisiä kirjastoja tarvitsen.

Päätin käyttää sivuston rakenteellisena pohjana Bootstrapia, joka on suosittu ilmainen HTML-, CSS- ja JavaScript-ohjelmistokehys (15). Bootstrapissa on lukuisia erilaisia CSS-pohjia ja JavaScript-komponentteja, jotka tekevät verkkosivun kehittämisestä huomattavasti nopeampaa. Bootstrap-kirjaston voi halutessaan räätälöidä täysin ja sisällyttää ainoastaan projektissa tarvittavat komponentit. Sivustoa saadaan kevyemmäksi ja latausaikoja pienennettyä, kun selaimen ei tarvitse ladata ylimääräisiä komponentteja, joita ei sivustolla käytetä. Projektissa käytin usein hyödyntämäni Bootstrapin mobiilnavigaatiokomponenttia ja -pohjia, sekä ruudukkopohjia (*grid*), joiden avulla sivuston sisältöelementit voidaan rakentaa ikään kuin rakennuspalikoista.

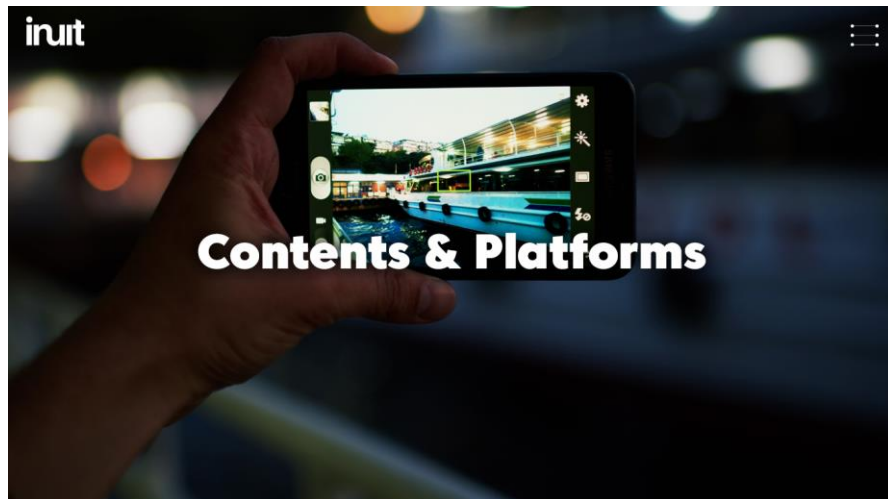
Ruudukkopohja perustuu 12-jakoiseen pohjaan, jossa sivusto jaetaan leveyssuunnassa 12:een yhtä leveään sarakkeeseen (16). HTML-elementille voidaan osoittaa mikä tahansa leveys väliltä 1/12 ja 12/12, mikä nopeuttaa työskentelyä huomattavasti, kun kehittäjän ei itse tarvitse laskea elementeille pikselileveyttä. Bootstrapin ruudukkopohjassa on myös vakiona mobiiliresponsiivisuus, jolloin elementtien leveyteen voidaan vaikuttaa CSS:n *media queryjen* breakpointeilla. Kuvassa 3 esitetään Bootstrapin ruudukkopohjan toiminnasta yksinkertainen esimerkki, jossa sivustolla on vierekkäin neljä yhtä leveää HTML-elementtiä. Jokaisen leveys on yli 1 200 pikseliä leveällä näytöllä 3/12 niitä ympäröivän säiliön (*container*) leveydestä. Kun selaimen kokoa pienennetään alle 1 200 pikselin leveyteen, HTML-elementtien leveys kasvaa leveyteen 6/12 säiliön leveydestä. Näkymässä on siis kaksi elementtiä vierekkäin ja niiden alla kaksi elementtiä vierekkäin. Kun selaimen kokoa pienennetään alle 767 pikselin leveyteen, eli mobiilinäkymään, jokainen elementti kasvaa täysleveäksi, eli leveyteen 12/12 säiliön leveydestä. Tällöin kaikki elementit ovat allekkain. Sama lopputulos olisi mahdollista toteuttaa CSS-koodilla, mutta Bootstrapia hyödyntämällä muutama HTML-luokkaluokitus (*class*) riittää.



Kuva 3. Esimerkki Bootstrapin ruudukkopohjan toiminnasta.

#### Uuden Inuits.fi-sivuston esittely

Kuvassa 4 esitellään Inuits.fi:n etusivun yläosa, joka pitää sisällään täysleveän 16:9-taustavideon, logon, navigaation avausnapin, videon päällä olevan animoidun tekstin ja video-osion alla olevan esittelytekstin.

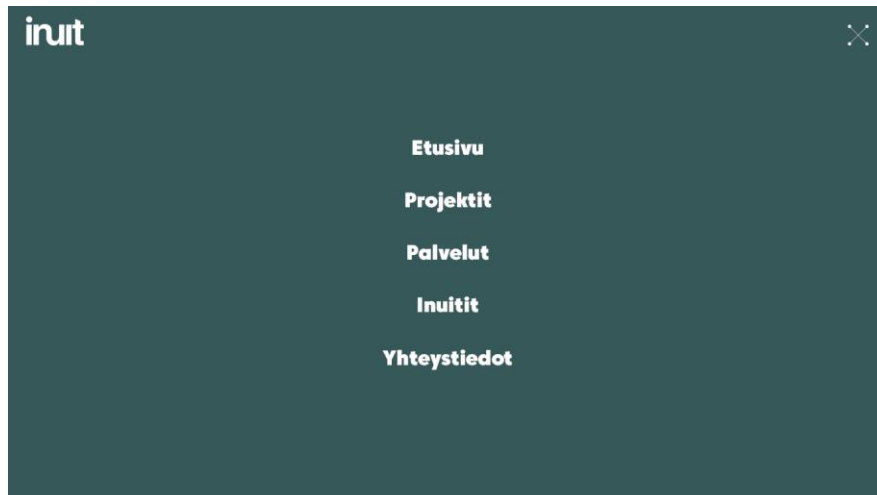


Markkinointikampanjoita, sisältöä, palveluita,  
 digitaalisia erikoistoteutuksia ja työkaluja.  
 Klassista mainontaa, tämän päivän  
 markkinointia ja tulevaisuuden next level -juttuja.  
 Luovaa ajattelua, freshiä teknologiaa ja  
 erikoisosaamista.



Kuva 4. Etusivun yläosa.

Taustavideo on tarkoitus rakentaa HTML video -elementillä, mikä tarkoittaa sitä, että videosta täytyy tehdä monta eri tiedostoversiota, sillä kaikki selaimet eivät tue muuten standardia mp4-videoformaattia (17). Videon 16:9-suhde on mobiiliresponsiivisuudelle haaste, sillä HTML-elementille ei ole mahdollista asettaa tiettyä suhdetta, jossa korkeus ja leveys pysyisivät samassa suhteessa riippumatta näytön koosta. Navigaation avausnappia painamalla navigaation on tarkoitus aueta kuvan 5 osoittamalla tavalla.



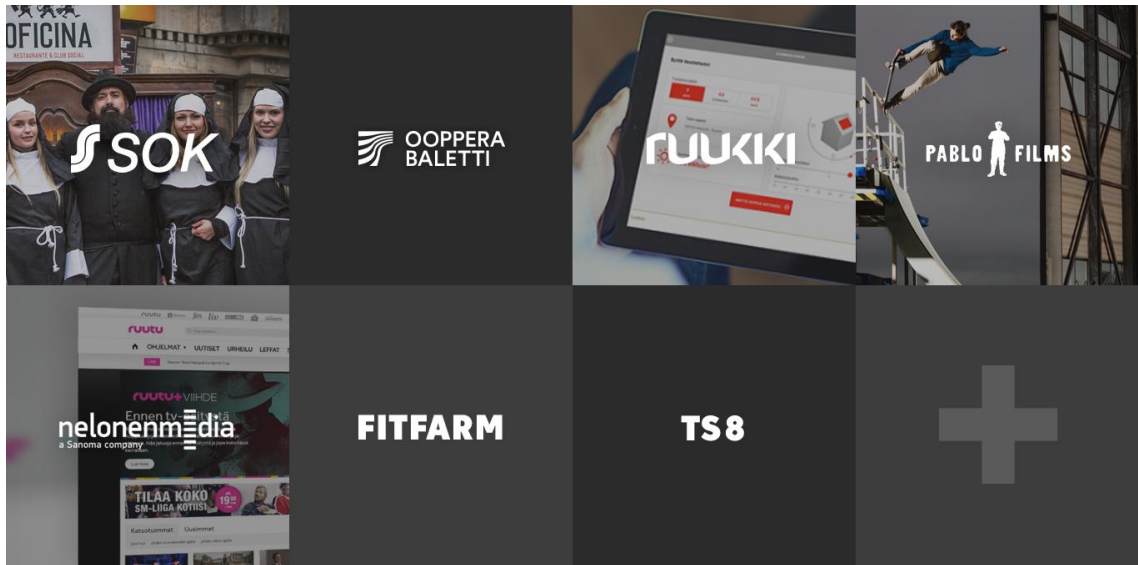
Markkinointikampanjoita, sisältöä, palveluita,  
digitaalisia erikoistoteutuksia ja työkaluja.  
Klassista mainontaa, tämän päivän  
markkinointia ja tulevaisuuden next level -juttuja.  
Luovaa ajattelua, freshiä teknologiaa ja  
erikoisosaamista.



Kuva 5. Etusivun yläosa, navigaatio avattuna.

Navigaatiopalkin tulee täyttää koko videon alue, mikä asettaa myös haasteita, jotka saattavat vaatia videon korkeuden tunnistamista JavaScriptillä. Navigaatiossa käytän Bootstrapin omia valmiita komponentteja projektiin sopivaksi muokattuna. Videon päällä oleva teksti vaihtuu ja tekstipaikalla on kolme eri virkettä. Tekstin sisääntuloanimaatiossa yksi kirjain tulee sisään kerrallaan, ikään kuin joku kirjottaisi tekstiä suoraan videon päälle. Video ja sen päällä olevat tekstit tehdään suoraan HTML:ään, minkä vuoksi niiden muuttamiseen SilverStripen hallintapaneelissa ei ole tarvetta. Videon alapuolella oleva esittelyteksti sen sijaan mahdollisesti vaihtuu tulevaisuudessa, minkä vuoksi se on järkevintä toteuttaa SilverStripen komponenteilla, jotta sitä voidaan muokata suoraan hallintapaneelistä.

Etusivun seuraava osio on lyhyt projektinostoseinä, johon nostetaan seitsemän tärkeintä tai ajankohtaisinta projektia kuvan 6 mukaan.



Kuva 6. Etusivun projektit-osio.

Jokainen projektinosto koostuu taustakuvasta, logosta ja niiden päällä olevasta linkistä, joka vie projektisivun tarkempaan projektiesittelyyn. Projektinostojen jälkeen sijaitsee +-nappi, joka vie projektien pääsivulle, josta kaikki projektit löytyvät samaan tapaan listattuna. Projektinostokuvan täytyy pysyä neliönä kaikilla näytöillä, ja pienemmillä näytöillä nostot järjestäytyvät ensin kaksi vierekkäin ja pienissä mobiilinäytöissä täysleveinä. Tässä hyödynnetään samaa logiikkaa kuin Bootstrapin taulukkokomponenteissa. Projektinostoja varten täytyy rakentaa SilverStripeen logiikka, jolla voidaan esitellä tiettyjä projekteja ja vaihtaa niitä jatkuvasti. Tämä tulee olemaan haaste, sillä en vielä tiedä, miten tämäntyyppinen ratkaisu on mahdollista toteuttaa SilverStripessa.

Projektien jälkeen tulee palveluesittelyosio, jossa kerrotaan lyhyesti kuvan 7 pohjalta Inuitsin pääpalveluista sekä esitellään joitain tärkeimpiä asiakkaita.



## Mitä teemme



## Asiakkaat



Suunnittelua markkinointikonseptista visuaalisuuden kautta tekniseen. Teknisiä toteutuksia ohjelmoinnista sisällönhallinnan kautta jatkokehitykseen. Verkkototeutuksia, printtimateriaaleja, videotuotantoja, äänisuunnittelua, AR-, RFID-, QR- ja NFC-sovelluksia sekä ihan uusia kokeiluja.



## Videotuotanto



## Äänistudio



## Inuit-expo



## Kätevä tapa esitellä ja aktivoida

Suunnittelemamme Expo-kosketusnäyttökioskit eri kokoisilla kosketusnäytöillä sopivat hyvin muun muassa tuotteiden ja palveluiden esittelyyn, kisoille ja asiakasaktivoineille messuilla, tapahtumissa, toimipisteissä sekä myymälöissä. Kosketusnäytöt ja ohjausyksiköt räätälöidään asiakaskohtaisesti. Kioskit ovat kolmiosaisia ja rakennettu helposti asennettaviksi. Tarjoamme kosketusnäytöille myös konseptisuunnittelua, sisällöntuotantoa ja ohjelmointipalveluita



Kuva 7. Etusivun palvelut-osio.

Palveluesittely jaetaan kahteen osaan: yläosa, jossa esitellään otsikot ”Mitä teemme” ja ”Asiakkaat”, sekä alaosa, jossa esitellään lyhyesti videotuotanto, äänistudio ja Inuit-Expo. Molemmassa osioissa esittelyteksti vaihtuu otsikkoa klikkaamalla. Kaikkien palve-



luiden tekstit eivät siis näy samaan aikaan, vaan käyttäjän täytyy klikata otsikoita nähdäkseen muita tekstejä. Pohjana voin käyttää Bootstrapin omaa dynaamista välilehtikomponenttia (*dynamic tabs*) kuvan 8 mukaan (18). Välilehtikomponentista saan perustoiminnallisuudet, jotka muokkaan CSS:llä graafisen suunnitelman näköiseksi.

## Dynamic Pills

To make the tabs toggleable, add the `data-toggle="pill"` attribute to each link. Then add a `.tab-pane` class with a unique ID for every tab and wrap them inside a div element with class `.tab-content`.

[Home](#) [Menu 1](#) [Menu 2](#) [Menu 3](#)

### Menu 1

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Kuva 8. Bootstrapin dynaamiset välilehdet (18).

Seuraava osio on kuvan 9 mukainen henkilöstöesittely, joka jatkaa samaa laatikkotemaakin kuin projektinostoseinä. Ideana on, että jokaisesta henkilöstä otetaan kaksi kuvaa: tavanomainen pääkuva ja ilmeeltään humoristisempi kuva. Perusnäkylässä henkilöistä näkyy pääkuva, nimi ja titteli. Kun käyttäjä vie hiiren henkilön päälle, vaihdetaan tilalle humoristisempi, sähköpostiosoite ja puhelinnumero. Henkilöille ei tehdä erillistä esittelysivua, mutta henkilöstöesittelyä varten täytyy silti rakentaa taustajärjestelmä SilverStrippen, jotta uusia henkilöitä voidaan lisätä ja vaihtaa kuvia tai muita yhteystietoja tarpeen mukaan.

<p><b>Aleksis Moisio</b> CEO aleksis.moisio@inuit.fi +358 (0)50 583 4466</p>	<p><b>Henrik Stürmer</b> Project Manager henrik.sturmer@inuit.fi +358 (0)50 518 0872</p>	<p><b>Markus Tamminen</b> Creative Director markus.tamminen@inuit.fi +358 (0)50 463 3250</p>	<p><b>Sampo Pihlaja</b> Technical Director sampo.pihlaja@inuit.fi +358 (0)40 568 6990</p>
<p><b>Aurinna Marstio</b> Concept Designer / Copywriter aurinna.marstio@inuit.fi +358 (0)50 367 6102</p>	<p><b>Pasi Kortström</b> Art Director pasi.kortstrom@inuit.fi +358 (0)40 742 2523</p>	<p><b>Arthur Kavander</b> Graphic Designer arthur.kavander@inuit.fi +358 (0)44 565 1331</p>	<p><b>Anu Lehto</b> Freelancer Designer anu.lehto@inuit.fi +358 (0)40 778 6783</p>
<p><b>Teemu Lahti</b> Developer teemu.lahti@inuit.fi +358 (0)44 075 6323</p>	<p><b>Lauri Laatikainen</b> Developer lauri.laatikainen@inuit.fi +358 (0)50 555 5508</p>	<p><b>Lewis Jackson</b> Developer lewis.jackson@inuit.fi +358 (0)</p>	<p><b>Petri Hoskonen</b> Developer petri.hoskonen@inuit.fi +358 (0)</p>

Kuva 9. Etusivun henkilöstöesittely.

Etusivun alaosassa annetaan yhteystiedot ja Inuitsin toimiston sijainti Google-kartalla. Yhteystietojen ohessa on myös jakonapit eri sosiaalisiin medioihin. Napit toteutetaan ai-noastaan linkkeinä, eikä niihin ole tarvetta sisällyttää jako-ominaisuuksia. Google-kartan rakennan itse Google Maps JavaScript API:n avulla, sillä Googlen yksinkertainen upo-tuskartta (*embed*) ei tue räätälöityjä paikkaosoittimia ja kartan väriystä (19).

Projektit-pääsivun ja projektiesittelysivun graafisen suunnitelmaan sain kuvamuodon si-jaan kirjallisena ohjeistuksena. Projektien pääsivulla esitellään projektit samaan tyyliin kuin etusivun projektinostoseinässä kuvassa 6, mutta tässä tapauksessa kaikki projektit näytetään. Projektinostoja varten käytetään samaa SilverStripe-pohjaa kuin etusivulla. Sivun yläosaan tulee samantyyppinen videotausta samoilla elementeillä kuin kuvassa 4, ja sivun alaosaan samat sosiaalisen median jakonapit kuin etusivun yhteystietojen alla kuvassa 10.



### Interactive Inuits Oy

Köydenpunojankatu 2 a D

00180 Helsinki, Finland

+358 (0)50 583 4466

info@inuit.fi

Y-tunnus: 22954980



Kuva 10. Etusivun yhteystiedot ja Google-kartta.

Projektisivulla yksittäiset projektit esitellään tarkemmin. Sivun yläosa tulee muuten samalla teemalla kuin kuvassa 4, mutta videon tilalla on staattinen täysleveä kuva. Kuvan päälle ei kuitenkaan tule samanlaista animoitua tekstiä, vaan kyseisen projektin asiakkaan logo ja avainsanoja projektiin liittyen. Yläosan jälkeen tulee artikkeliosio projektin kuvausta varten. SilverStripessa on vakiona otsikko- ja artikkelialueet, minkä vuoksi niitä ei tarvitse itse tehdä. Artikkelin jälkeen on tarkoitus rakentaa Vimeo- ja YouTube-video-paikkoja sekä kuvagalleria. Kuvagalleriassa esitellään ensin kuvista pienet versiot, ja kuvaa painamalla aukeaa kuva sivun päälle täyteen kokoon. Kuvagalleriaa varten päätin käyttää valmista jQuery Lightgallery -kirjastoa, joka on mobiiliresponsiivinen, modulaarinen ja helposti muokattava ja joka tukee mobiililaitteiden kosketuseleitä kuvien vaihtamiseen (20).

### 3.4 Toteutus

Kattavan esikartoituksen jälkeen pyysin Lead Developer Teemu Lahtea luomaan sivustoa varten uuden tyhjän SilverStripe-projektin BitBucketiin, jota käytämme Git-projektien (*repositories*) hallintaan. Tyhjä projekti sisälsi ainoastaan esiasennetun SilverStripen tie-

tokantoinen sekä Gulpin ja NPM:n. Gulp sisältää SASS-esiprosessorin, joka mahdollistaa SCSS-koodin käyttämisen ja lokaalin testisivun automaattisen uudelleenlatauksen. NPM toimii paketinhallintatyökaluna, minkä vuoksi asensin kaikki tarvitsemani kirjastot sen kautta. Lokaalin palvelimen pystyitin hyödyntäen Vagrantia ja VirtualBoxia.

## Sivupohjien ja -mallien luominen

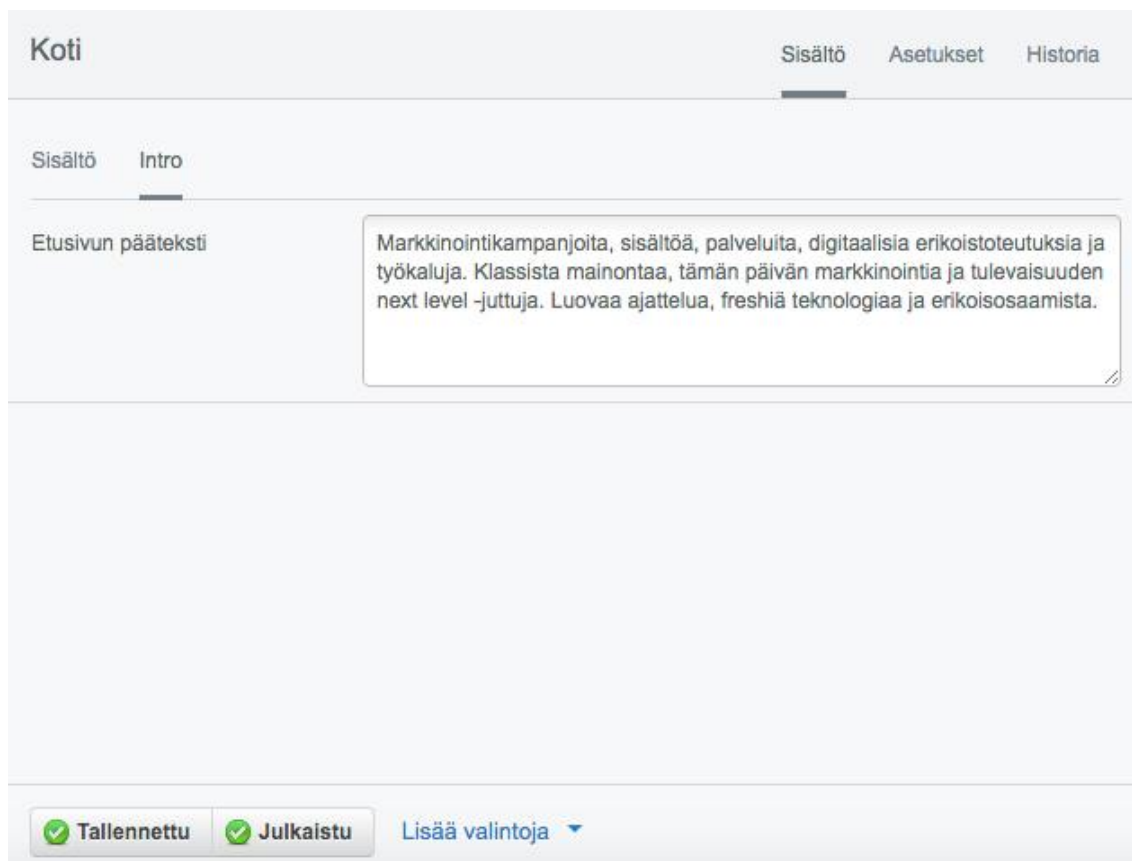
Aloitin luomalla kolme eri sivutyyppiä SilverStripeen: etusivu, projektien pääsivu ja yksittäisen projektin esittelysivu. Sivupyyppien lisääminen on yksinkertaista, ja se vaatii vain kaksi tiedostoa sivutyyppiä kohden. Ensin loin sivumallit themes-hakemistoon nimillä `FrontPage.ss` (etusivu), `ProjectHolderPage.ss` (projektien pääsivu) ja `ProjectItemPage.ss` (yksittäisen projektin esittelysivu). Sivumallien yhdistämistä SilverStripeen varten loin sivupohjat `mysite/code`-hakemistoon samoilla nimillä, mutta `php`-tiedostopäätteillä. Esimerkkikoodissa 1 näkyy yksinkertainen sivupohja, jolla sivumallin saa yhdistettyä SilverStripen hallintapuoleen. On tärkeää, että sivumalli, sivupohja ja sivupohjassa käytetty luokka `class` ovat samalla nimellä, sillä muissa tapauksissa SilverStripe ei osaa yhdistää niitä. Loin `FrontPage.php:n` lisäksi myös `ProjectHolderPage.php:n` ja `ProjectItemPage.php:n`. Kun sivumallit ja -pohjat oli luotu, pystyin lisäämään niiden pohjalta sivuja sivustolle. Tämä vaatii kuitenkin sivupohjien lisäämisen tietokantaan sekä SilverStripen välimuistin tyhjentämisen. Uusien kenttien lisääminen tietokantaan onnistuu automaattisesti selaimen osoiterivillä, osoitteella `http://inuits.dev/dev/build`, jossa `http://inuits.dev` on lokaalin testisivun osoite ja `/dev/build` lisää uudet kentät tietokantaan automaattisesti kertoen, mikäli `php`-koodista löytyy virheitä. SilverStripen välimuistin tyhjentäminen onnistuu osoiterivillä osoitteessa `http://inuits.dev?flush=all`. Sivupohjaan, eli `PHP`-tiedostoon, muutoksia tehtäessä täytyy aina päivittää tietokanta, ja sivumalleihin, eli `SS`-tiedostoihin, muutoksia tehtäessä täytyy tyhjentää välimuisti.

Kaikki sivupohjat ja -mallit laajentavat pääluokkaa `Page`, joka toimii säiliönä kaikelle muulle sisällölle. `Page.ss`:sä määritellään sivun `Head`-elementin sisältö ja muita kaikilla sivuilla toistuvia osioita, kuten `Footer`-elementti. Inuitsin verkkosivuprojektissa `Head`-elementin lisäksi kaikilla sivuilla toistuu ainoastaan `Header`-elementti, joka on logon ja navigaation sisältävä sivun yläosa. Sisältöalue, eli osa, johon tulee muiden sivupohjien sisällöt, määritellään `Page.ss`:sä SilverStripen omalla muuttujalla `$Layout`, jonka tilalle SilverStripe osaa automaattisesti laittaa näkyvillä olevan sivun sisällöt. `Page.php`:ssa voidaan määritellä yleisiä sivupohjamuuttujia ja tietokantakenttiä. Tässä tapauksessa jätin `Page.php:n` tyhjäksi, sillä mitään yleisiä tietokantakenttiä ei tarvittu.

## Etusivu

Etusivun rakentamisen aloitin loogisesti ylhäältä alas, aloittaen siis videosta. Taustaku-  
vavideota varten sain MP4-videon, jonka konvertoin WebM-videoksi mahdollisimman  
laajaa selaintukea varten. Halusin eriyttää videon etusivusta, jotta sitä voisi käyttää myös  
projektien pääsivulla, joten tein sitä varten oman tiedoston Video.ss. Tällä tavoin voin  
lisätä videon haluamaani paikkaan yksinkertaisesti SilverStripen komennolla `<% in-  
clude Video %>`. Toteutin videon perinteisenä HTML5-videona säätimet piilotettuna,  
automaattisesti itseään toistavana ja äänettömänä. Videota siis toistetaan taustalla koko  
ajan, eikä käyttäjän ole tarkoitus päästä pysäyttämään sitä. Mobiiliresponsiivisuutta var-  
ten lisäsin Div-elementin Video-elementin ympärille, jolle määritin täyden leveyden ja  
korkeuden hieman poikkeavalla määritelmällä `padding-bottom: 56.25%`, joka var-  
mistaa, että elementin lapsielementit pysyvät aina suhteessa 16:9. Div-elementin sisällä  
olevalle video-elementille asetin määrittymisen `position: absolute`, joka mahdollistaa  
elementin kellumisen ulomman elementin sisällä. Tämän jälkeen asetin videon täyttä-  
mään kaiken käytössään olevan alueen, jolloin video on aina samassa suhteessa kor-  
keus- ja leveyssuunnassa. Videon päälle tulevat tekstit animoin hyödyntäen CSS:n ani-  
mationia ja @keyframeja.

Video-osion jälkeen oli vuorossa etusivun esittelytekstin paikan asettaminen, joka on vi-  
deon alla. Loin yksinkertaisen Bootstrap-rakenteen, jota käytän muissakin tekstiosioissa  
sivustolla. Uloimman elementin luokka on *container*, joka keskittää sen sisällön auto-  
maattisesti. *Container*-luokan leveys on yli 1 200 pikseliä leveillä näytöillä 1 170 pikseliä,  
mutta sen koko pienenee pienemmillä näytöillä. Seuraava elementti on luokaltaan *row*,  
joka määrittää uuden rivin. Sisäin Div-elementti sisältää useamman luokan, ja niillä mää-  
ritellään sisällön leveyttä entisestään. Luokka `col-md-8` kertoo, että elementin leveys  
on 8 saraketta yli 992 pikseliä leveillä näytöillä ja luokka `col-sm-10` määrittää 10 sa-  
rakkeen leveyden näyttökokoilla 768px–992px. Luokka `col-md-offset-2` siirtää ele-  
menttiä kahden sarakkeen verran oikealle, eli tässä tapauksessa keskittää elementin. P-  
elementin sisälle tein SilverStripen muuttujan `$Text1`. Tämän muuttujan tekstin liittämi-  
nen hallintapaneeliin onnistui luomalla tietokantamuuttujassa `$db` uuden tietokantaken-  
tän luokalla *Text*, joka mahdollistaa raakatekstin syöttämisen. Lisäsin äsken määrittele-  
mäni tietokantakenttä `Text1:n` SilverStripen hallintapaneeliin välilehdelle ”Intro” otsikolla  
”Etusivun pääteksti”. Kuvassa 11 on näkymä etusivun hallintapaneelistä, johon lisäsin  
etusivun pääotsikon. Tekstikenttään kirjoittamalla ja tallentamalla teksti ilmestyy sivus-  
tolle näkyviin sille määritellyyn paikkaan.



Kuva 11. Etusivun pääteksti SilverStripen hallintapaneelissa.

Projektien nosto-osion jätin etusivun rakennusvaiheessa kokonaan välistä, sillä siihen tulen käyttämään samaa tyylitystä kuin projektien pääsivulla, jonka rakennan myöhemmin.

Seuraavana oli etusivun palvelut-osio, joka muuttui hieman kuvasta 7, sillä totesimme, ettei ole tarvetta kertoa palveluista niin laajasti vaan pyrimme tyypistämään sisällön yhteen osioon otsikolla "Mitä teemme" kuvan 12 mukaisesti. Kala- ja muut kuvakkeet toteutin staattisina kuvina HTML:ssä, sillä niitä ei ole tarvetta muuttaa myöhemmin. Kuvan 12 otsikkoa ja tekstiosiota varten rakensin saman Bootstrap-rakenteen kuin edellä, mutta tietokantamuuttujat tein eri tavalla.



## Mitä teemme

Olemme nopealiikkeinen ja tuloshakuinen digimarkkinointitoimisto. Luomme toimivia kampanjakokonaisuuksia, erottuvia ilmeitä ja luovia konsepteja. Sivustot ja sähköiset palvelut toteutamme ideasta käyttäjystävälliseen viimeistelyyn.

Käyttöliittymäsuunnittelu, ohjelmointi, sosiaalisen median sovellukset ja digitaalisten ratkaisujen sisällönsuunnittelu ovat ydinosiamme. Videotuotantomme tekee niin brändivideoita, taktista videomainontaa kuin luovia produktioita. Meiltä löytyy erikoisosaamista myös haastaviin ja kokeellisiin markkinointiprojekteihin. Testaamme uusia teknologioita ja keksimme tuoreita tapoja käyttää niitä luoden asiakkaillemme lisäarvoa.

Kuva 12. Etusivun ”Mitä teemme” -osio.

Tietokantakentän `Text`-luokka ei sisällä HTML-tageja, minkä vuoksi laitoin sen tässä tapauksessa `P`-elementin sisään, jotta saadaan oikeaoppista HTML-koodia. Sama pätee muuttujaan `$Title2`, jota varten loin tietokantakentän luokalla `Varchar(255)` tiedostoon `FrontPage.php`, joka mahdollistaa enintään 255-merkkisen tekstin lisäämisen. Otsikolle tuo pituus on sopiva, mutta se tarvitsee silti vielä `H2`-elementin ympärilleen ollakseen oikeaoppista HTML-koodia. Tämä ei kuitenkaan päde ”Mitä teemme” -otsikon alla olevaan tekstialueeseen, johon haluttiin vaikuttaa enemmän. ”Mitä teemme” -otsikon alla olevan tekstin muuttujaa `$Text2` varten loin tietokantakentän luokalla `HTMLText` ja hallintapaneeliin kentän `HTMLEditorField` tekstin muokkaamista varten. Tämä mahdollistaa WYSIWYG-editorin käyttämisen tekstin muokkaamiseen hallintapuolella. Koska luokan `HTMLText` ulostulo on valmista HTML-koodia, ei sitä tarvitse sulkea minkään HTML-elementin sisään.

Ihmiset-osio, eli henkilöstöesittely, vaati paljon aikaa, sillä sain paljon vaikutusvaltaa sen toteutukseen. Lähdin rakentamaan kuvan 9 osiota hallintapaneelin tarpeiden mukaan. Yksi henkilö tarvitsee kaksi kuvaa, tavanomaisen ja humoristisen kuvan, nimen, tehtävänimikkeen, sähköpostiosoitteen ja puhelinnumeron. Lähdin rakentamaan ensin back-end-puolta, sillä se tuntui loogisimmalta, jotta pystyisin paremmin hahmottamaan kokonaisuuden. Henkilöstöesittelyssä tulee olemaan useita henkilöitä, joita täytyy pystyä lisäämään ja poistamaan. Lisäksi henkilöiden tietoja tulee pystyä muuttamaan. Näiden toiminnollisuuksien varmistamiseksi loin henkilöä varten abstraktin luokan `DataObject`,

joka mahdollistaa tietokannan relaatiot eri luokkien välillä. Tietokantaan määrittelin kentät *Name*, *JobTitle*, *Email*, *Phone* ja *PhoneLink* sekä kuvat *Image1* ja *Image2*. Tietokantamuuttujan `$db` sisällä määrittelin tavalliset `Varchar(255)`-tekstikentät ja muuttujassa `$has_one` `Person DataObject`-luokan relaation etusivuun ja kahteen kuvaan. `$has_one` luo *1-to-1*-relaation tietokantaobjektien välille, jossa jokainen henkilö on suhteessa etusivuun. Etusivulla voi kuitenkin olla useampi henkilö, mikä määritellään muuttujassa `$has_many`, joka luo *1-to-many*-relaation. Henkilön tietokantakenttien vieminen hallintapaneeliin onnistuu osittain automaattisesti, sillä muuttujan `$db` tekstikentät `SilverStripe` siirtää suoraan hallintapaneeliin. `SilverStripe` luo tietokantakentät kuvia varten myös automaattisesti, mutta kuvan hallinnoimista varten sille täytyy luoda uusi latauskenttä (`UploadField`). Määritin molemmille kuville omat latauskentät ja sallitut tiedostomuodot, kuvien maksimimäärän ja kohteiden tallennuskansion. Henkilöiden lisäämistä varten loin etusivun hallintapaneeliin uuden välilehden "Persons", johon loin luokan `GridFieldConfig_RelationEditor`, joka mahdollistaa uusien relaatio-objektien lisäämisen listanäkymässä kuvan 13 mukaan.

Koti Sisältö Asetukset Historia

Sisältö Intro **Persons**

[+ Lisää Person](#)  [Linkitä olemassaoleva](#)

Person		Näytä 1–14 / 14
Nimi	Titteli	
Aleksis Moisio	CEO	
Kalle Alanen	Account Director	
Henrik Stürmer	Project Manager	
Markus Tamminen	Creative Director	
Sampo Pihlaja	Technical Director	

Kuva 13. Etusivun Persons-välilehti ja GridField-editoriluokka.

Kuvassa 14 on näkymä uuden Person-objektin luomisesta hallintapaneelissa.



Koti /  
Uusi Person

Name

Job Title

Email

Phone

Phone Link

Front Page Koti

Pääkuva

Liitä tiedosto (tallentuu polkuun: /inuit-images)

Toinen kuva

Liitä tiedosto (tallentuu polkuun: /inuit-images)

Kuva 14. Person-objektin luominen.

Työntekijöiden lisäämisen toteutin etusivulle SilverStripen loop-tagilla, joka mahdollistaa kaikkien samantyyppisten objektien iteroinnin ja toimii siis hyvin pitkälti samalla periaatteella kuin JavaScriptin for-loop. Määrittelin etusivun PHP-koodissa DataObject-luokan "Person" ja niille yhteisen muuttujan `$Persons`, jotta minun olisi mahdollista iteroida koko henkilöstö kerralla. Muuttuja `$Persons` on määritetty ainoastaan etusivulla, eikä sitä voida käyttää millään muulla sivulla. Visuaalisen puolen toteutin luomalla yhden div-elementin henkilön säiliöelementiksi, jolle annoin leveyden `width: 25%` ja korkeuden `height: 25vw`, joka tekee elementistä lähes neliön mallisen. Koko henkilöstöesittely on täysleveyttä, eli henkilöitä on perusnäkyvässä neljä vierekkäin ja jokaisen henkilön säiliöelementin leveys on neljäsosa koko sivun leveydestä. CSS3-mittayksikkö "vw", eli *viewport width*, mittaa koko sivun leveyttä: 100vw on koko sivun leveys ja 25vw neljäsosa siitä. Tätä mittayksikköä on mahdollista käyttää myös elementin korkeudelle, jolloin elementin korkeus on 25 % koko sivun leveydestä. Mittayksikkönä vw ei ole kuitenkaan

täysin absoluuttinen, sillä se laskee leveyteen mukaan myös vierityspalkin. Sivun leveyden lisäksi on mahdollista myös mitata sivun korkeutta mittayksiköllä *vh*, eli *viewport height*, jolloin lasketaan sivun näkyvää korkeutta eikä koko sivun pituutta.

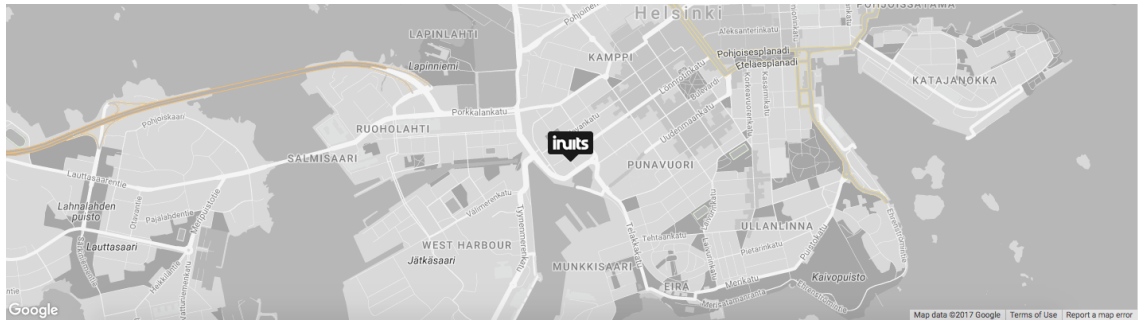
Uloimman säiliöelementin sisään laitoin kaksi säiliön täyttävää *div*-elementtiä CSS-määrittämisellä `position: absolute`, joista ensimmäisessä on taustakuvana henkilön taivonomainen kuva ja toisessa humoristinen kuva, jotka vaihtavat paikkaa, kun hiiren vie elementin päälle. Kaikista henkilöistä ei löytynyt kuvia, ja siksi korvasin kuvan tummalla taustavärillä. Tein *if*-lauseen, jolla tarkastellaan, löytyykö henkilöltä peruskuvaa `$Image1`, jos löytyy, näytetään kuva taustakuvana, ja jos ei, näytetään pelkkä taustaväri. Saman logiikan tein myös humoristisempaa kuvaa `$Image2` varten. Mikäli kuva löytyy, se määritellään elementin *style*-attribuutissa CSS:llä käyttäen muuttujan `$Image1` tai `$Image2` parametria *URL*, joka palauttaa kuvan absoluuttisen *URL*-osoitteen. SilverStripe-kuvien käyttäminen suoraan CSS-tiedostossa ei onnistu, sillä SilverStripe ei käänne CSS-tiedostoja ja näin ollen myös muuttujat jäävät kääntämättä. Tästä syystä jouduin lisäämään CSS-määrittäykset suoraan elementin *style*-attribuutin sisään. Loin *if*-lauseen myös puhelinnumeron tarkistusta varten, sillä kaikkien henkilöiden puhelinnumeroita ei haluttu julkisiksi.

Etusivun palveluosio päätettiin korvata otsikolla ”Mitä teemme” ja yhdellä tekstiosuella, mutta tärkeimmät asiakkaat haluttiin silti esitellä, joten tein staattisen kuvallista asiakkaiden logoista kuvan 15 mukaan. Arthur Kavander rajasi kaikki logot kuvasuhteeseen 1:1. Niitä varten en kokenut tarpeelliseksi tehdä omaa tietokantaosiota, sillä uusien asiakkaiden lisäämiseen ei ole tarvetta.



Kuva 15. Lopputulos etusivun Asiakkaat-osiosta.

Google-kartan toteutin hyödyntäen Google Mapsin rajapintaa. Kartasta oli tavoitteena tehdä mahdollisimman yksinkertainen ja sivuston tyyliin sopiva, joten poistin siitä kaikki ylimääräiset karttakuvakkeet ja tekstit. Kartan saturaation määritin negatiiviseksi, jolloin sain aikaiseksi harmaavalkoisen efektin. Määritin kartan osoittamaan Inuitsin toimiston koordinaatteihin ja lisäsin keskikohdan päälle oman Inuits-logolla varustetun kuvakkeen. Piilotin kaikki kartan UI-elementit ja poistin kartasta vieritysominaisuuden, jotta sivuston vieritys ei häiriintyisi. Lopputulos on nähtävissä kuvassa 16.



Kuva 16. Lopputulos etusivun Google-kartasta.

## Projektit

Projektien esittelysivua ja etusivun projektinostoja varten rakensin ensin yksittäisten projektien esittelysivun, joita ilman en olisi saanut haettua sisältöjä muille sivuille. Loin jokaiselle projektille kolme kuvakenttää: `PhotoMain`, eli neliön mallinen pikkukuva nostoja varten, `Logo`, eli asiakasyrityksen logo, jota käytetään nostoissa ja projektisivun yläosassa ja `Bg`, eli projektisivun suuri taustakuva. Projektisivun yläosan rakensin kuvan 2 mukaan, ja sitä varten loin edellä mainitut taustakuva- ja logopaikat sekä avainsanakentän `Keywords`.

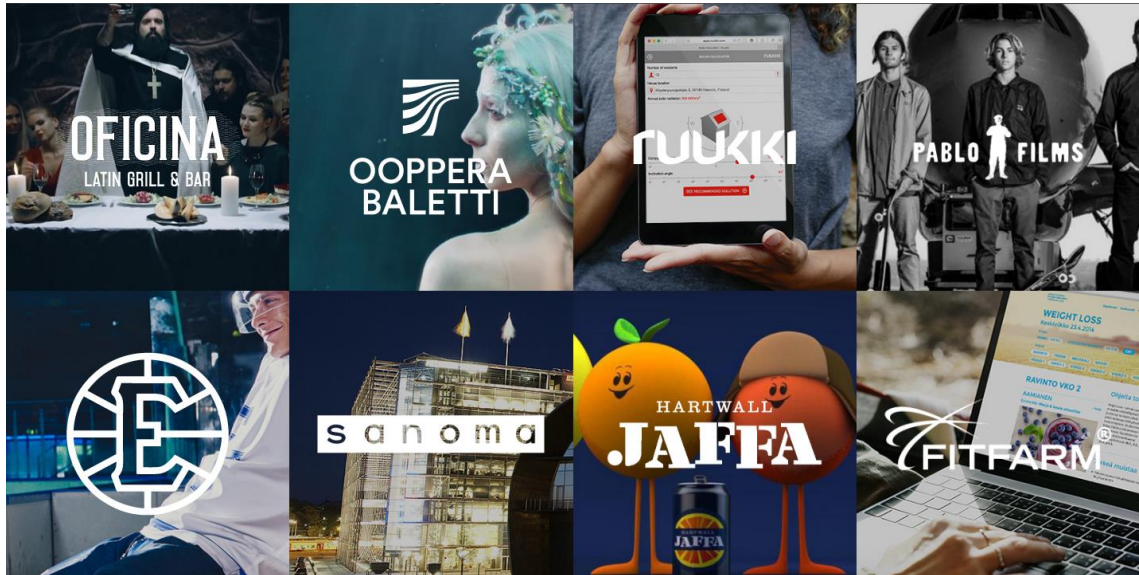
Avainsanakentän tyyppinä käytin `Varchar(255)`, ja kentässä voidaan esitellä joitain avainsanoja projektiin liittyen. Projektisivun pääkuvaosion alle tein otsikkopaikan `Heading` tyyppillä `Varchar(255)` ja sisältötekstialueen `MainContent` tyyppillä `HTMLText`, jotta tekstiin voi sisällyttää HTML-koodia. Koska etusivulla haluttiin näyttää vain tietyt projektit, loin nostettavia projekteja varten totuusarvomuuttujan `ShowOnHomepage` (`Boolean`) tietokantaan ja sitä vastaavan valintaruudun (`checkbox`) hallintapaneeliin. Tarkoituksena on siis näyttää etusivun nostoissa ainoastaan projektit, joilla totuusarvomuuttujan `ShowOnHomepage` arvo on tosi.

Jokaisessa projektissa haluttiin esitellä myös joitain kuvia ja mahdollisesti myös videoita, joita varten loin uuden abstraktin `DataObject`-luokan `Media`. `Media` voi olla joko kuva tai YouTube- tai Vimeo-video, ja sille voidaan määrittää lyhyen esittelytekstin sivulla näytettäväksi. Tein medialle tietokantatekstikentät `Name`, jota käytetään ainoastaan hallintapaneelissa median tunnisteena, `VideoURL`, jossa voidaan määrittellä videon URL-osoite, ja `Description`, johon voidaan syöttää lyhyt esittelyteksti. Kaikki tekstikentät toteutin tyyppinä `Varchar(255)`. Loin lisäksi totuusarvomuuuttajat `IsVideo`, jolla määritetään median tyyppi (kuva vai video), ja `IsVimeo`, jolla määritetään videon tyyppi (YouTube vai Vimeo).

Tein myös relaation luokkien `Media` ja `ProjectItemPage` välille. Median projektisivulle lisäämistä varten loin luokan `GridField` uudelle välilehdelle ”Gallery” projektisivun hallintapaneeliin. Galleriakuvien esittämistä varten sivustolla käytin JavaScript-kirjasto `light-gallery`, joka luo automaattisesti kuvista thumbnail-kuvat, joita klikkaamalla kuva aukeaa alkuperäiseen kokoonsa koko ruudun kokoisena tummana peittona. Koko ruudun näkymä sisältää nuolinapit, joiden avulla käyttäjä voi siirtyä kuvasta toiseen. `Lightgallery` tukee myös mobiiliin kosketuseleitä. Asensin `lightgallery`n npm:llä.

Projektisivut loin `SilverStrip`essa projektit-sivun alasuiviksi, jolloin projektisivujen URL-osoite on muotoa `http://inuits.fi/projektit/projektinimi`. `SilverStrip`essä on oma logiikkansa alasuivujen esittämiseen niiden yläsuivulla muuttujalla `$AllChildren`. Koska kaikki projektisivut ovat projektit-pääsuivun alasuivuja, pystyin iteroimaan kaikkien projektien tietoja projektit-pääsuivulla. Tein samantyyppisen loop-rakenteen kuin henkilöstöesittelyssä. Loin yksittäisen projektin esittämistä varten `div`-elementin, jolle annoin saman koon kuin henkilöstöesittelyssä. Lisäksi loin `if`-lauseen, joka tarkistaa, onko projektille annettu neliönmallinen projektinostokuva `$PhotoMain`. Jos kuva on annettu, se näytetään `div`-elementin taustakuvana, muussa tapauksessa näytetään ainoastaan yksivärinen tausta. Kuvan määritin samaan tapaan kuin henkilöstöesittelyssä, mutta lisäsin siihen määrittelyn `SetHeight(500)`, jolla `SilverStripe` luo kuvasta automaattisesti pienennetyn version, tässä tapauksessa korkeuden 500 pikseliä mukaan. Määrittely `SetHeight()` ei muuta kuvasuhdetta, vaan skaalaa kuvan korkeuteen 500 pikseliä. Tämä on hyvä tapa varmistaa kuvien optimointi sivustolla, sillä liian isot kuvat hidastavat sivun latausaikoja. `SilverStrip`essä on mahdollista automaattisesti myös rajata kuvia ja muuttaa kokoa venyttämällä. Tein `if`-lauseen myös logon tarkistamiseksi, jolloin tarvittaessa näy-

tetään logon tilalla projektin pääotsikko. Projektit ryhmittelin sivustolle ominaiseen tapaan neljän riveihin kuvan 17 mukaan. Mobiililaitteissa projektikuvia on vierekkäin kaksi tai yksi, riippuen laitteen koosta.



Kuva 17. Projektit-sivun projektiesittelyosion lopputulos.

Etusivun projektinostoseinää varten tein samanlaisen näkymän kuin kuva 17 projektit-sivulla. Projektien tietoja en kuitenkaan voinut etusivulle suoraan hakea, vaan sitä varten loin funktion `Projects()`, joka hakee ja palauttaa sivupohjan `ProjectHolderPage` tiedot. Pystyin nyt iteroimaan muuttujalla `$Projects` kaikkien `ProjectHolderPage`-sivutyypin läpi etusivulla, joita tässä tapauksessa oli kuitenkin vain yksi. Projektinostoseinän rakenteen tein muuten samalla tavalla kuin projektit-sivulla, mutta lisäsin siihen `if`-lauseen, jossa tarkistetaan totuusarvomuttujan `ShowOnHomepage` arvo. Kaikki projektit, joille muuttujan `ShowOnHomepage` arvoksi on määritetty `true`, näkyvät etusivulla.

#### 4 Projektin lopputulos ja sivustoverailu

Työn tuloksena oli mielestäni erittäin näyttävä ja hyvin toimiva sivusto, joka esittelee hyvin Inuitsin pääperiaatteita, eli yksinkertaista, näyttävää ja teknisesti modernia ja kaikilla päätelaitteilla toimivaa. Sivustoprojekti eteni melko hitaasti, ja tein sitä syyskuusta 2016 tammikuuhun 2017, minkä jälkeen sivusto julkaistiin. Inuitsin toimitusaikataulu vastaavanlaajuisille projekteille on keskimäärin 4–8 viikkoa, ja alkuperäisestä aikataulusta oltiin myöhässä jo ennen kuin projekti edes saapui työpöydälleni. Sivusto haluttiin toteuttaa

mahdollisimman hyvin, mutta sen suunnitteluun käytettiin turhan paljon aikaa. Muutoksia sivustoon tehtiin paljon myös ohjelmointivaiheen aikana, mikä johti osaltani kymmeniin ylimää räisiin työtunteihin. Esimerkiksi etusivun tekstiosiot (kuva 7) päätettiin toteuttaa huomattavasti yksinkertaisemmin (kuva 12) ja videon päällä olevan tekstin (kuva 4) sisältö ja animointi muutettiin kokonaan.

Insinööriyössä tehdyn kotisivuprojektin tarkoituksena oli korvata yrityksen nykyiset kotisivut, joten looginen tapa tutkia projektin onnistuneisuutta oli verrata uutta ja vanhaa sivustoja. Vertailun toteutin kyselynä, johon osallistui sekä Inuitsin henkilökuntaa että muita media-alan ammattilaisia. Kuvasta 18 selviää eri ryhmien osuus kyselyssä. Mielestäni jakauma on sopiva ottaen huomioon projektin tarkoituksen, joka oli rakentaa teknisesti toimiva sivusto. Päätin teettää kyselyn myös muilla media-alan asiantuntijoilla, sillä Inuitsin henkilöstöstä noin puolet on teknisesti vähemmän valveutuneita ja halusin painottaa kyselyssä juuri teknistä puolta. Koko kyselyn otanta oli vain kymmenen henkilöä, mikä jäi itseäni hieman harmittamaan, sillä olisin toivonut enemmän osallistujia. Otannan pienyydestä huolimatta uskoisin, että tulokset ovat silti hyvin lähellä totuutta.



Kuva 18. Kotisivuvertailuun osallistuneiden jakautuneisuus.

Molempia sivustoja varten tein kyselyn identtisillä kysymyksillä, jotta vertailu olisi helpompaa. Kysely koostui kolmesta osuudesta: kaksi vapaata tekstikenttää ja yksi väittämäosio, joka sisälsi seitsemän erilaista väittämää. Ensimmäinen kysymys oli ”Millainen yleisfiilis sivustosta tulee?”. Tämä oli avoin tekstikenttä, johon toivoin vastaajan kertovan sivustosta yleisesti. Tarkoituksena ei ollut tästä kentästä kerätä mitattavaa dataa, vaan herättää vastaaja miettimään sivustoja hieman yleisesti. Teemat, jotka vastauksissa esiintyivät, olivat seuraavan mukaiset.

Uusi sivusto:

- selkeä
- ”fresh”
- tyylikäs / visuaalinen
- pelkistetty

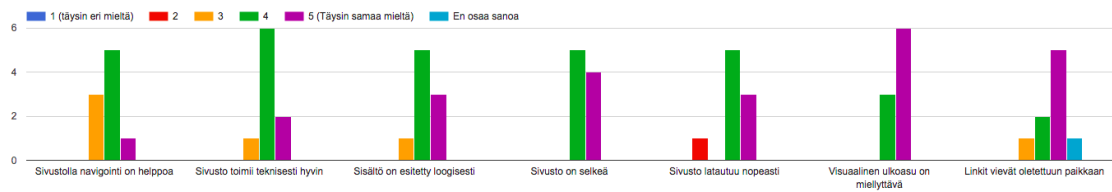
Vanha sivusto:

- ulkoasullisesti hyvä
- käytännöllisyys ontuva
- sekava / sotkuinen
- ihan kiva

Tästä voidaan vetää jo joitain johtopäätöksiä, joiden mukaan uusi sivusto on ensivaikutelmaltaan näyttävämpi, raikkaampi ja selkeämpi.

Väittämäosio koostui seitsemästä eri väittämästä, joihin vastaajan tuli valita vaihtoehtojen 1 (täysin eri mieltä) ja 5 (täysin samaa mieltä) väliltä omaa mielipidettään kuvaavin. Vastausvaihtoehtona oli myös ”En osaa sanoa”, joka ei onneksi ollut suosittu vastaus. Väittämät käsittelivät sivuston teknistä toimivuutta, selkeyttä, latausaikoja, visuaalisuutta, linkkejä ja sivustolla navigoimista. Tällaisen kyselyn tekeminen oli minulle uutta, joten luin joitain artikkeleita ja päädyin tekemään kyselyn Utah State Universityn oppien mukaan (21), sillä niiden avulla uskoin saavani käyttökelpoista dataa.

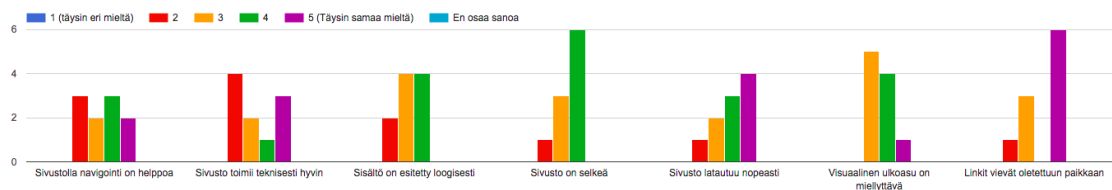
Kuvassa 19 on nähtävillä vastausten jakautuminen uutta sivustoa koskeviin väittämiin. Väittämäkohtaisista keskiarvoista voidaan vetää johtopäätös, että sivusto suoriutui vähintään erinomaisesti kaikilla muilla alueilla paitsi navigoinnin helppoudessa, joka jäi hiekan alle kouluarvosanan 4. Parhaiten uusi sivusto pärjasi visuaalisen ulkoasun miellyttävyydessä, joka oli yksi tärkeimmistä tavoitteista. Myös yleiset toiminnallisuudet on vastausten mukaan toteutettu erinomaisesti, joka oli toinen tärkeä tavoite.



3,777...	4,111...	4,222...	4,444...	4,111...	4,666...	4,5
----------	----------	----------	----------	----------	----------	-----

Kuva 19. Uuden sivuston kyselytulosten yhteenveto.

Kuvasta 20 selviää vanhan sivuston vastausten jakautuminen. Vastaukset tukevat yllättävän hyvin edeltäviä avoimia vastauksia, joiden mukaan sivusto toimii kaikin puolin hyvin, mutta parannettavaa on. Tätä tukevat alhaiset keskiarvot sivuston teknisessä toimivuudessa ja sisällön esittelyn loogisuudessa. Parhaiten vanha sivusto pärjasi latausajoissa ja linkkien loogisuudessa.



3,4	3,3	3,2	3,5	4	3,6	4,1
-----	-----	-----	-----	---	-----	-----

Kuva 20. Vanhan sivuston kyselytulosten yhteenveto.

Yhteenvetona molemmista väittämätulosten jakautumisesta voidaan todeta, että uusi sivusto on kaikilta osin parempi kuin vanha sivusto. Pienin ero oli sivustojen latausajoissa, minkä parantaminen ei kuitenkaan ollut projektin tavoitteena. Suurimmat erot olivat sivuston teknisessä toimivuudessa, selkeydessä ja visuaalisuudessa, eli juuri niillä osaluilla, joiden parantaminen oli projektissa tärkeää.

Väittämäosion jälkeen oli avoin tekstikenttä otsikolla ”Yleisiä huomioita ja havaittuja virheitä”, ja sen tarkoituksena oli kerätä dataa ainoastaan uuden sivuston mahdollisista ongelmista. Mielestäni tämä oli jälkeinpäin ajateltuna erittäin hyvä kysymys, sillä sen avulla löysin pari kriittistä ongelmaa: navigaatio ei auennut kaikilla käyttäjillä ja käytettävyyden kannalta navigaation avausnapin sijoittaminen ainoastaan sivun yläreunaan oli



huono päätös. Tästä on nähtävissä suora seuraus uuden sivuston huonolle tulokselle väittämökohdassa ”sivustolla navigointi on helppoa”.

Navigaation avausnapin koodiin oli jäänyt pieni virhe, joka esti sen toimimisen joillain laitteilla. Lisäämällä napin `addEventListener`in `useCapture = false` sain toimivuuden taatua paremmin. Ulkoasullisista syistä navigaationappi on sijoitettu ainoastaan sivun yläosaan, eikä se siis monien muiden sivustojen tapaan seuraa mukana käyttäjän selatessa sivua alaspäin. Nopeuttaakseni sivuston alaosasta pääsyä takaisin navigaatioon lisäsin sivun oikeaan alareunaan monista WordPress-teemoistakin tutun ”takaisin ylös” -napin, joka seuraa käyttäjää koko ajan oikeassa alareunassa. Lisäsin nappiin animaation, jotta sain aikaiseksi nopean vieritysefektin takaisin sivun yläosaan. Uskoisin, että tällä tavalla sivuston käytettävyys kohentui huomattavasti ja mikäli teettäisin kyselyn uudestaan, uusi sivusto pärjäisi huomattavasti paremmin navigoinnin helppoudessa.

Kyselyn aikaan uusi sivusto oli Inuitsin testipalvelimella, joka on jonkin verran hitaampi kuin palvelin, jossa vanha sivusto oli. Tämä saattoi vaikuttaa latausaikoihin hieman, mutta sitä ei ollut tarpeen tutkia enempää, sillä sivuston latausaikojen parantaminen ei ollut osana projektin tavoitteita.

## 5 Yhteenveto

Insinöörityössä tein ensin kattavan esiselvityksen käytetyimmistä avoimen lähdekoodin sisällönhallintajärjestelmistä ja vertailin niitä asiakasyrityksen käyttämään SilverStripen. Sisällönhallintajärjestelmien vertailu oli osittain hieman hankalaa, sillä jotkin niiden osista poikkeavat toisistaan paljon.

Esiselvityksen jälkeen tutustuin tarkemmin SilverStripen käyttöön ja sen ominaisuuksiin, tavoitteena oppia tarpeeksi, jotta voisin toteuttaa verkkosivuprojektin asiakasyritykselle Inuits. Esiselvitys oli tärkeä ja informatiivinen osa oppimisprosessia, mutta SilverStripen toiminnot iskostuivat mieleeni paremmin vasta sivuston rakennusvaiheessa. Insinöörityön päätarkoituksena oli opetella SilverStripen käyttöä ja näiden oppien avulla rakentaa teknisesti toimiva ja näyttävä sivusto.

Sivustoprojekti oli myöhässä aikataulusta jo heti aluksi minusta riippumattomista syistä, ja siihen käytettiin muutenkin mielestäni turhan paljon aikaa. Useiden kommenttikierrosten takia jouduin tekemään ylimääräistä työtä useita tunteja, mikä olisi voitu välttää paremmalla vaatimusten määrittelyllä ennen ohjelmoinnin aloittamista. Projektin aikana vastaan tuli joitain kompastuskiviä, joita en saanut itse suoraan ratkottua, mutta onneksi Lead Developer Teemu Lahti pystyi ohjaamaan minua oikeaan suuntaan. Lopputuloksena sivusto oli vaatimusten mukainen, ja siihen oltiin Inuitsilla erittäin tyytyväisiä. Kaikkiin tavoitteisiin päästiin lukuun ottamatta alkuperäistä julkaisupäivää, ja lopputulos oli vanhaan sivustoon nähden parempi kaikilla osa-alueilla. Sivustovertilu tuki ja todisti tätä päätelmää. Sivusto julkaistiin tammikuussa 2017 osoitteessa <http://www.inuits.fi>, ja se korvasi Inuitsin vanhan sivuston.

Vaikka sivusto on nyt insinööriyöni osalta ohi, jatkuu sen kehitys kuitenkin aktiivisesti. Sivustolle on tarkoitus seuraavaksi toteuttaa markkinoinnin automaatiota, jonka avulla Inuits voi paremmin hyödyntää sivustolla käyviä potentiaalisia asiakkaita, sekä blogi, jonka tarkoituksena on hankkia lisää kävijöitä sivustolle mielenkiintoisten artikkeleiden perässä. Suunnitelmissa on myös panostaa referenssien laatuun ja markkinointiin sosiaalisessa mediassa.

Henkilökohtaisena päämääränäni oli oppia SilverStripen käyttöä mahdollisimman hyvin, jotta voisin jatkossa hyödyntää saamiani oppeja Inuitsin asiakasprojekteissa. Tähän päämäärään pääsin mielestäni erittäin hyvin, ja olen toteuttanut jo kaksi sivustoa eri asiakkaille SilverStripen pohjalta.

## Lähteet

- 1 Rouse, Margaret. 2013. Enterprise Content Management (ECM). Verkkodokumentti. <<http://searchwindowsserver.techtarget.com/definition/enterprise-content-management-ECM>>. Päivitetty syyskuu 2013. Luettu 26.9.2016.
- 2 Wordpress vs. Drupal vs. Joomla vs. Silverstripe – Choosing the right CMS in 2015. 2015. Verkkodokumentti. Notliveyet.com. <<https://notliveyet.com/blog/wordpress-vs-drupal-vs-joomla-vs-silverstripe-choosing-the-right-cms-in-2015/>>. Luettu 1.4.2017.
- 3 Meining, Robert. 2017. WordPress vs Joomla vs Drupal. Verkkodokumentti. <<https://websitesetup.org/cms-comparison-wordpress-vs-joomla-drupal/>>. Luettu 2.4.2017.
- 4 Volpe, Hugh. 2015. SilverStripe vs WordPress. Verkkodokumentti <<https://www.silverstripe.org/blog/silverstripe-vs-wordpress/>>. Luettu 2.4.2017.
- 5 Magnusson, Sigurd. 2010. SilverStripe CMS the first ever open source web app to become Microsoft Certified. Verkkodokumentti. <<https://www.silverstripe.org/blog/silverstripe-cms-the-first-ever-open-source-web-app-to-become-microsoft-certified/>>. Luettu 26.9.2016.
- 6 Awards. 2015. Verkkodokumentti. SilverStripe. <<https://www.silverstripe.com/who-we-are/awards/>>. Luettu 26.9.2016.
- 7 SilverStripe. 2017. Verkkodokumentti. Stack Overflow. <<http://stackoverflow.com/tags/silverstripe/info>>. Luettu 26.9.2016.
- 8 Craig, David. 2014. AngularJS Powered by SilverStripe. Verkkodokumentti. <<https://www.silverstripe.org/blog/angularjs-powered-by-silverstripe/>>. Luettu 26.9.2016.
- 9 Carlino, Aaron. 2016. Cutting through the noise: Why SilverStripe 4 will use ReactJS. 2016. Verkkodokumentti. <<https://www.silverstripe.org/blog/cutting-through-the-noise-why-silverstripe-4-will-use-reactjs/>>. Luettu 26.9.2016.
- 10 Getting Started. 2015. Verkkodokumentti. SilverStripe. <[https://docs.silverstripe.org/en/3.4/getting\\_started/](https://docs.silverstripe.org/en/3.4/getting_started/)>. Luettu 26.9.2016.
- 11 Building a basic site. 2015. Verkkodokumentti. SilverStripe. <[https://docs.silverstripe.org/en/3.4/tutorials/building\\_a\\_basic\\_site/](https://docs.silverstripe.org/en/3.4/tutorials/building_a_basic_site/)>. Luettu 26.9.2016.
- 12 Carlino, Aaron. 2014. Creating Your First Theme. Verkkodokumentti. <<https://www.silverstripe.org/learn/lessons/creating-your-first-theme>>. Luettu 26.9.2016.

- 13 Getting Started. Verkkodokumentti. Composer. <<https://getcomposer.org/doc/00-intro.md>>. Luettu 27.9.2016.
- 14 Wood, Kevin. 2016. 7 Reasons Why You Should Choose WordPress For Your Website. Verkkodokumentti. <<http://www.hostgator.com/blog/reasons-why-wordpress-website/>>. Luettu 4.1.2017.
- 15 Bootstrap 3 Tutorial. 2016. Verkkodokumentti. W3Schools. <<http://www.w3schools.com/bootstrap/>>. Luettu 29.1.2017.
- 16 Grid System. 2017. Verkkodokumentti. Bootstrap. <<https://v4-alpha.getbootstrap.com/layout/grid>>. Luettu 4.1.2017
- 17 Media formats supported by the HTML audio and video elements. 2017. Verkkodokumentti. Mozilla Developer Network. <[https://developer.mozilla.org/en-US/docs/Web/HTML/Supported\\_media\\_formats](https://developer.mozilla.org/en-US/docs/Web/HTML/Supported_media_formats)>. Luettu 18.2.2017.
- 18 Bootstrap tabs and pills. 2017. Verkkodokumentti. W3Schools. <[https://www.w3schools.com/bootstrap/bootstrap\\_tabs\\_pills.asp](https://www.w3schools.com/bootstrap/bootstrap_tabs_pills.asp)>. Luettu 20.2.2017.
- 19 Customizing a Google Map: Custom markers. 2017. Verkkodokumentti. Google Developers. <<https://developers.google.com/maps/documentation/javascript/custom-markers>>. Luettu 20.2.2017.
- 20 JQuery lightgallery. 2017. Verkkodokumentti. Sachinchoolur. <<https://sachinchoolur.github.io/lightGallery/>>. Luettu 20.2.2017.
- 21 Website usability Questionnaire. 2008. Verkkodokumentti. Utah State University. <<http://ocw.usu.edu/instructional-technology-learning-sciences/producing-distance-education-resources/resource30.html>>. Luettu 22.1.2017