

Opinnäytetyö (AMK)

Tietotekniikan koulutusohjelma

Hyvinvointiteknologia

2016

Aleksi Korpelainen

VALOKUVAUSSIVUSTON TOTEUTUS JA KÄYTTÖÖNOTTO DJANGOLLA

Aleksi Korpelainen

VALOKUVAUSSIVUSTON TOTEUTUS JA KÄYTTÖÖNOTTO DJANGOLLA

Työn tarkoituksena oli toteuttaa ja ottaa käyttöön toimiva valokuvaussivusto asiakkaalle. Sivuston tuli olla visuaalisesti näyttävä ja sen tuli sisältää hallintapaneeli sivuston sisällönmuokkausta varten. Lisäksi asiakkaalle tehtiin käyttökoulutus sivustonhallintaa varten.

Työ toteutettiin Pythonia käyttävän Django-www-sovelluskehityksen avulla. Projektissa käytiin läpi koko Django-sovelluksen elinkaari aina julkaisuun asti. Lisäksi sivuston julkaisemista varten konfiguroitiin virtuaalipalvelin. Sivustoa varten luotiin myös front end.

Lopputuloksena on toimiva sekä visuaalisesti näyttävä sivusto. Sivusto sisältää myös hallintapaneelin, jonka avulla asiakas pystyy itse hallitsemaan sivustolla näytettäviä kuvia sekä lukemaan palauteomakkeen vastauksia ilman teknistä tietämystä.

ASIASANAT:

Julkaisujärjestelmät, JavaScript, CSS, Python, MySQL, WWW-sivut

Aleksi Korpelainen

DEVELOPMENT AND DEPLOYMENT OF A DJANGO-POWERED PHOTOGRAPHY WEBSITE

The purpose of this Bachelor's thesis was to develop and deploy a functional photography website for a client. The website should be visually impressive and it should include a control panel for editing content on the website. Additionally, the client would undergo training for administrating the website.

The project was developed using the Django web framework, which uses Python. During the project the whole life cycle of a Django project including its publishing was examined and implemented. Additionally, a virtual server was configured to host the project. A front end was also created for the project.

The result is a working and a visually impressive website. The website also includes a control panel that the client, without having technical knowledge, can use to administrate images shown on the page or to read messages sent via the contact form.

KEYWORDS:

Content management systems, JavaScript, CSS, Python, MySQL, WWW-pages

SISÄLTÖ

KÄYTETYT LYHENTEET TAI SANASTO	6
1 JOHDANTO	7
2 KÄYTETYT TEKNOLOGIAT	9
2.1 HTML5, CSS3, Javascript ja jQuery	9
2.2 Javascript-tiedostot	10
2.3 Ohjelmistokehykset	10
2.4 Foundation 5	11
2.5 Python ja Django	11
3 SIVUSTON RAKENNE JA TOTEUTUS	13
3.1 Django-projektin aloittaminen	13
3.2 Julkaisujärjestelmän käyttöönotto Django-projektissa	13
3.3 Ulkoasu	15
3.4 Osien toteutus	17
3.5 Rakenteen toteutus	21
SIVUSTON JULKAISU	23
3.6 Virtuaalipalvelimen hankinta	23
3.7 Virtuaalipalvelimen konfigurointi ja testaus	23
4 LOPPUKÄYTTÄJÄN OHJEISTUS JA KOULUTUS	25
5 TULOKSET JA JOHTOPÄÄTÖKSET	26
LÄHTEET	27

LIITTEET

Liite 1. sakarimajantie.materiaali.fi-ohje sivustonhallintaan.

KUVAT

Kuva 1. Django-sovelluksen visuaalinen hallintakäyttöliittymä.	15
Kuva 2. Toimeksiantajan antama visuaalinen suunnitelma sivustoa varten työpöytänäkymässä.	16
Kuva 3. Toimeksiantajan toimittama visuaalinen suunnitelma sivuston mobiilisivusta.	16
Kuva 4. Ruudunkaappaus sivuston pääsivulta työpöytänäkymässä.	17
Kuva 5. Ruudunkaappaus sivuston pääsivulta mobiilinäkymässä.	18
Kuva 6. Ruudunkaappaus sivuston galleriasivulta työpöytänäkymässä.	19
Kuva 7. Ruudunkaappaus galleriasivusta mobiilinäkymässä.	19
Kuva 8. Ruudunkaappaus tilanteesta, jossa sivuston galleriasivulla olevaa kuvaa on klikattu työpöytänäkymässä.	20
Kuva 9. Ruudunkaappaus sisällysivusta, jonka päätavoite on näyttää sivuston käyttäjille tekstiä.	21
Kuva 10. Projektin template-tiedostokansion sisältö. Tekstin sininen ja vihreä väriyty liittyvät versionhallintaan.	21
Kuva 11. Projektin Javascript-tiedostokansion sisältö.	22

KOODIT

Koodi 1. HTML-dokumentissa esiintyvän div-elementin valitseminen perinteisen Javascriptin avulla.	9
Koodi 2. HTML-dokumentissa esiintyvän div-elementin valitseminen JQueryn avulla.	9
Koodi 3. Django-projektin settings.py-tiedostoon määritellyjä parametreja tietokantayhteyden luomista varten.	14
Koodi 4. base.html-tiedoston body-tagin sisältämä koodi, joka on sekoitus HTML- ja Djangoan template-koodia.	22

KOMENNOT

Komento 1. Djangoan asennus.	12
Komento 2. Uuden Django-projektin luonti.	13
Komento 3. Tietokantataulujen luonti Djangoilla.	14
Komento 4. Migraatioiden suorittaminen tietokantaan.	14

KÄYTETYT LYHENTEET TAI SANASTO

Back end	Back end kuvaa ohjelmistorajapintaa, joka toimii ohjelmiston taustalla loppukäyttäjän ulottumattomissa.
CSS	Cascading Style Sheets. Ohjelmointikieli, jolla määritellään HTML-koodin tuottamien elementtien ulkoasu.
DOM	Document Object Model. DOM kuvaa esimerkiksi HTML-dokumentin elementtien hierarkiaa.
Front end	Front end kuvaa ohjelmistorajapintaa, josta ohjelmiston loppukäyttäjä on riippuvainen. Front end kuvastaa siis sitä ohjelmiston puoliskoa, jonka loppukäyttäjä näkee.
Full stack	Front end ja back end -rajapinnat yhdessä.
HTML	HyperText Markup Language. Ohjelmointikieli, jota käytetään internet-sivujen sisällön rakentamiseen.
Template-kieli	Ohjelmointia nopeuttava ohjelmointikieli, jota käytetään jonkin tietyn ohjelmiston sisäisenä kielenä. Toimiakseen se vaatii muita läsnäolevia dokumentteja, ja tyypillisesti sitä käytetään projektin sisäisesti.

1 JOHDANTO

Projektin tarkoituksena on toteuttaa responsiivinen verkkosivusto valokuvaajan portfolioikäyttöön. Sivuston avulla valokuvaajana toimiva asiakas pystyy lataamaan ottamiinsa valokuvia internetiin. Sivusto toimii myös valokuvaajan esittelysivustona. Toimeksiantajana opinnäytetyössä on Perjantai Markkinointiviestintä. Sivuston perusrakenne on toteutettu HTML5-, CSS- ja Javascript-ohjelmointikielillä.

Www-suunnittelussa responsiivisuudella tarkoitetaan sitä, että verkkosivuston sisältö näytetään kullekin laitteelle sopivassa muodossa. Sillä siis pyritään luomaan käyttäjän laitteen näyttöön sopiva käyttöliittymä riippumatta laitteesta. Responsiivinen verkkosivu toimii oikein ja näyttää hyvältä sekä työpöytäkoneella että mobiililaitteella. [1]

Verkkosivuston tulee sisältää osiot asiakkaan tiedoille sekä valokuville. Toimeksiantajan vaatimus verkkosivustolle on näyttävä ulkoasu. Tiettyjen verkkosivuston osien on myös oltava asiakkaan muokattavissa. Valokuvien lisäämiseen ja poistamiseen sekä tiettyjen sisältöalueiden muokkaamiseen on pystyttävä käyttämään julkaisujärjestelmää.

Kun tehdään www-sivustoa, jota asiakas päivittää itse, on sen pohjalle tärkeää luoda julkaisujärjestelmä. Julkaisujärjestelmän avulla sivuston sisällön tuottaminen muuttuu pelkän koodikielen käytöstä visuaalisempaan muotoon. Julkaisujärjestelmän käyttöönotto on helppoa eikä sen käytön opettelu vaadi paljoa aikaa. Toisin sanoen julkaisujärjestelmän käyttö sopii sellaiselle sivuston ylläpitäjälle, joka ei tunne sivustonhallinnan teknistä puolta. [2]

Projektin alussa julkaisujärjestelmän toteuttamiseen valitaan julkaisujärjestelmä kolmen eri vaihtoehdon väliltä: Django-www-ohjelmistokehys, Wordpress-julkaisujärjestelmä sekä CMS Made Simple -julkaisujärjestelmä (CMSMS). Wordpressiin on saatavilla suuren käyttäjäkunnan myötä eniten valmiita ratkaisuja, CMSMS taas on joissakin toimeksiantajan vanhoissa projekteissa käyttämä julkaisujärjestelmä. Projektissa käytetään kuitenkin Djangoa, sillä toimeksiantaja on valinnut sen aikaisemmin useaan projektiin www-ohjelmistokehykseksi ja projektin edetessä toimeksiantajalta on mahdollista saada apua Django käytössä. Valintaa tukee myös se, että lähes kaikki toimeksiantajan uudet projektit tehdään Djangolla. Www-sovellusohjelmistokehys Django tuo mukanaan valmiin julkaisujärjestelmän.

Toimivaan Django-asennukseen on julkaisujärjestelmän perusversio implementoitu sekä kytketty päälle valmiiksi. Luvussa 2 käydään läpi projektin kannalta tärkeimmät teknologiat. Luvussa 3 perehdytään tarkemmin julkaisujärjestelmän käyttöönottoon sekä tarvittavien lisäominaisuuksien asennukseen.

2 KÄYTETYT TEKNOLOGIAT

2.1 HTML5, CSS3, Javascript ja jQuery

HTML5 on viimeisin standardoitu versio HTML- eli Hypertext Markup Language -ohjelmointikielestä, jota käytetään Internet-sivustojen luomiseen [3]. CSS eli Cascading Style Sheets -ohjelmointikieli on luotu HTML-kielellä tehdyn koodin visuaalisen ilmeen määrittelemistä varten. Viimeisin standardoitu versio CSS:stä on CSS3. CSS:llä voidaan HTML-dokumentin rakentamaan sisältöön määritellä esimerkiksi värejä, fontteja, elementtien kokoja ja marginaaleja sekä taustakuvia. [4]

Javascript on skriptaus- eli komentosarjakieli, joka on suunniteltu pääasiassa interaktiivisuuden lisäämiseen verkkosivuihin. Javascript voidaan sulauttaa sivustoilla suoraan HTML-koodin sekaan. Yleensä sivustojen Javascript-komentojen määrittelemät toiminnot tapahtuvat esimerkiksi sivunlatausten, tiettyjen elementtien klikkauksien tai näppäimistön käytön yhteydessä. [5]

Javascript on oliopohjainen ja se tukee muutamia valmiiksi sisäänrakennettuja olioita. Ohjelmoijan on mahdollista myös luoda tai poistaa omia olioitaan. Javascript on myös tulkattu (eng. interpreted) ohjelmointikieli eikä sitä tarvitse ennen suorittamista kääntää (eng. compile) järjestelmän ymmärtämäksi kieleksi. [5]

jQuery on kevyt Javascript-kirjasto, joka mahdollistaa Javascriptin kirjoittamisen tiiviimmässä muodossa. JQueryyn tarkoitus on helpottaa Javascriptin luomista sekä käyttämistä verkkosivustoilla. JQueryyn avulla dokumentissa navigointi, DOM (eng. Document Object Model) -elementtien eli käytännössä HTML-elementtien valitseminen, animointi sekä tapahtumien hallinnointi helpottuu huomattavasti perinteiseen Javascriptiin verrattuna. Kaikki projektissa käytettävät Javascript-tiedostot hyödyntävät JQuerya.

Esimerkkinä koodissa 1 on esitetty HTML-dokumentissa esiintyvän div-elementin valitseminen perinteisen Javascriptin avulla:

```
document.getElementsByTagName("div"); (1)
```

Sama toiminto JQuerylla toteutettuna koodissa 2:

```
$('.div'); (2)
```

Koodia 2 voidaan käyttää, jos halutaan kohdistaa toimintoja kaikkiin HTML-dokumentin div-elementteihin. Molemmat esimerkit päättyvät samaan lopputulokseen. Esimerkki havainnollistaa, kuinka paljon JQueryn käyttö vähentää tarvittavan koodin määrää verrattuna perinteiseen Javascriptiin.

2.2 Javascript-tiedostot

Projektissa Javascript-tiedostoja käytetään useita. Sivuston toiminnan kannalta tärkeät lisäosatiedostot ovat Wookmark, Lightbox sekä scrollReveal.

Wookmark uudelleenjärjestää halutun elementin sisältämät elementit niin, että kuvaelementtien väliin jäävät tyhjät alueet jäävät ainoastaan galleriaelementin alareunaan. Wookmarkille on mahdollista antaa parametreja, jotka muuttavat järjestettävien elementtien käyttäytymistä ja ulkonäköä. [6]

Sivustolla olevaa valokuvaa klikattaessa aukeaa käyttäjän selainikkunaan valokuvasta suurempi versio. Suurennettujen valokuvien päälle luodaan painikkeet helpottamaan valokuvien selaamista. Tämä toiminnallisuus saadaan aikaan käyttäen Lightbox-nimistä Javascript-lisäosaa. [7]

ScrollReveal-lisäosa mahdollistaa DOM-elementtien helpon animoinnin. ScrollRevealia käytetään sivuston front end -puolella visuaalisen näytävyyden lisäämiseksi. [8]

2.3 Ohjelmistokehykset

Ohjelmistokehyksiä käyttämällä ohjelmiston tuotantoa voidaan nopeuttaa ja helpottaa huomattavasti. Ohjelmistokehyksellä tarkoitetaan valmiiksi ohjelmoitua kirjastoa, runkoa, jonka pohjalle ohjelmisto rakennetaan. Ohjelmistokehys hoitaa esimerkiksi välttämättömät, monta kertaa toistettavat asiat, jotka hidastavat ohjelmiston tai ohjelman suunnittelua. [9]

Ennen ohjelmistokehyksen käyttämistä sen perusominaisuudet ja -toiminnot on opeteltava. Esimerkiksi Foundation 5 -ohjelmistokehys on helppokäyttöinen, mikäli käyttäjä osaa www-ohjelmoinnin perusteet. Toisaalta esimerkiksi Django vaatii käyttäjältään huomattavasti enemmän kokemusta www-ohjelmoinnista.

2.4 Foundation 5

Zurb-yhtiön luoma Foundation 5 on ohjelmistokehys responsiivisen front end -puolen rakentamiseen sivustolle. Foundation 5 -ohjelmistokehystä käytettäessä sivuston rakenne tukeutuu valmiiksi luotuihin luokkiin. Foundation 5 -projektissa HTML-elementeille annetaan valmiiksi ohjelmistokehyksessä määritellyjä CSS-luokkia. Peruseriaatteena on, että HTML:n div-elementit jaetaan riveihin ja niiden sisältämät div-elementit sarakkeisiin. Foundation 5 luo elementeille automaattisesti responsiivisuuden.

Foundation 5 käyttää tyylien määrittelyyn Sass-kieltä, joka on CSS:n tyyppinen tyylliteltyyn tarkoitettu kieli. Sass on laajennettu versio CSS-kielestä, joka mahdollistaa HTML-elementtien tyylien määrittelyn helpommin ja nopeammin kuin perinteistä CSS-kieltä käytettäessä. Jotta selaimet voivat lukea tuotettuja tyylimääritelmiä, Sass muutetaan CSS:ksi käyttämällä esimerkiksi esikäntäjää. Tässä projektissa esikäntäjänä käytetään Compass-nimistä sovellusta. [10]

2.5 Python ja Django

Python on oliopohjainen ohjelmointikieli. Python-koodia on mahdollista suorittaa monilla Unix-järjestelmillä, Macilla sekä Windows-käyttöjärjestelmillä Windows 2000:sta eteenpäin. [11]

Django on Python-ohjelmointikieleen pohjautuva www-sovellusten kirjoittamiseen tarkoitettu ohjelmistokehys, jota käyttävät maailmanlaajuiset yritykset kuten Instagram, Pinterest ja Mozilla. Django rakennettiin alun perin täyttämään nopealla aikataululla www-sisältöä tuottavan World Onlinen tarpeet. [12]

Djangon oppimiskynnys on suurempi verrattuna esimerkiksi Foundation 5 -ohjelmistokehykseen, sillä toisin kuin Foundation 5:ssä, Django on ns. full stack -ohjelmistokehys. Yksin Django ja Django-projektiin asennettavien moduulien avulla voidaan siis luoda www-sovelluksen loppukäyttäjäräjäpinnan lisäksi myös palvelintoiminnallisuus. [13]

Suurin osa minkä tahansa olemassaolevan Django-projektin tiedostoista on kirjoitettu Python-ohjelmointikielellä. Lisäksi Django käyttää omaa template-kieltään, joka toimii välittäjänä tietokannan ja sovelluksen visuaalisen ulkoasun välillä. Osa näistä tiedostoista tulee Django-asennuksen yhteydessä valmiina, mutta osan voi myös luoda itse. [14]

Django-projektia aloitettaessa on projektin isännöivälle tietokoneelle oltava asennettuna Python-ohjelmointikieli, paikallinen tietokantapalvelin sekä itse Django-ohjelmistokehys. Python-ohjelmointikieli on asennettava ensin, sillä Djangon tiedostot asentuvat järjestelmän Python-kansioon. Djangon asentamiseksi suositellaan asennettavaksi pip-nimistä Python-lisäosaa, joka vastaa kaikkien muiden tulevaisuudessa asennettavien lisäosien hallinnasta. Pip tulee yleensä mukana Python-asennuksen yhteydessä. Kun järjestelmään on asennettu Python ja pip, voidaan komentorivillä suorittaa komento 1, joka asentaa Djangon järjestelmään. [15]

pip install Django

(1)

Django-asennuksen yhteydessä projektiin luodaan valmiiksi perusta julkaisujärjestelmälle. Jos se jätetään koskemattomaksi, sen avulla voidaan hallita ainoastaan julkaisujärjestelmän käyttäjiä sekä käyttäjäryhmiä. Projektin julkaisujärjestelmään on kuitenkin mahdollista luoda uusia alueita, joilla voidaan hallita www-sivuston sisältöä. [16]

3 SIVUSTON RAKENNE JA TOTEUTUS

3.1 Django-projektin aloittaminen

Uusi Django-projekti luodaan navigoimalla komentorivillä haluttuun kansioon ja sen jälkeen suorittamalla komennossa 2 havainnollistettu komento.

```
django-admin startproject mysite (2)
```

Komennon osa "mysite" korvataan omalla nimellä. Komennon tuloksena syntyy Django-projekti, joka sisältää tarpeelliset tiedostot Django-projektin aloittamiseksi. [17]

3.2 Julkaisujärjestelmän käyttöönotto Django-projektissa

Tämän projektin julkaisujärjestelmään luodaan galleriaosio, jolla voidaan hallita sivuston kuvagallerioita. Julkaisujärjestelmään luodaan myös osio näyttämään sivuston palautelomakkeen avulla lähetettyjä palautteita.

Toimiakseen oikein Django tarvitsee moduuleita, jotka asennetaan paketteina järjestelmän Python-kansioon. Kaikki projektit eivät kuitenkaan tarvitse samoja moduuleita tai käyttävät eri versioita samasta moduulista. Tämän takia tässä projektissa otetaan käyttöön työkalu virtualenv, joka mahdollistaa eristettyjen Python-ympäristöjen luomisen ja käyttämisen projektikohtaisesti. Projektin vaatimat riippuvuudet voidaan siis kansioda ja tarvittaessa ottaa käyttöön tai poistaa käytöstä. Lisäksi eristetyn ympäristön käyttö mahdollistaa sen, että järjestelmän varsinaisen Python-asennuksen yhteyteen asennetut moduulit on helpompi pitää järjestyksessä. Nimi virtualenv tulee englannin kielen sanoista "virtual environment". [18]

Toimiakseen Django tarvitsee yhteyden sekä oikeudet tietokantaan. Tässä projektissa käytetään MySQL-tietokantaa. Tuotantovaiheessa projektin tietokanta toimii paikallisesti localhost-palvelimella, ja projektin julkaisuvaiheessa se luodaan uudelleen varsinaiselle palvelimelle. Koodissa 3 Django-projektin settings.py-tiedostossa määritellään, miten Django-projekti ottaa yhteyden tietokantaan. Koodin 3 tietokantakäyttäjän oikea salasana on korvattu tekstillä "xxxxxxx" tietoturvasyistä. Mikäli kohdat "HOST" ja "PORT" jätetään tyhjiksi, kertoo tämä Djangolle, että sovellus käyttää paikallista palvelinta. [19]

```
DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.mysql',

        'NAME': 'sakarimajantie',

        'USER': 'sakarimajantie',

        'PASSWORD': 'xxxxxxx',

        'HOST': '',

        'PORT': '',

    }

}
```

Koodi 3. Django-projektin settings.py-tiedostoon määriteltyjä parametreja tietokantayhteyden luomista varten.

Tietokantataulut luodaan Djangolla suorittamalla komento 3 siinä projektin kansiossa, jossa manage.py-tiedosto sijaitsee. Yleensä kyseinen kansio sijaitsee projektin juurikansiossa.

python manage.py makemigrations (3)

Tämä komento luo projektiin luoduista malleista (eng. model) migraatitiedostot, jotka sisältävät tietokantakomentoja. Tämän jälkeen suoritetaan komento 4, joka suorittaa migraatioiden sisältämät komennot tietokantaan.

python manage.py migrate (4)

Mikäli malleja luodaan tai poistetaan, migraatitiedoston luominen ja suorittaminen tekee tarvittavat muutokset automaattisesti tietokantaan. [20]

Djangossa on mahdollista luoda visuaalinen käyttöliittymä halutuista tietokantamalleista. Tällöin mallin määrittelemiin tietokantasarakkeisiin voidaan lisätä tai niistä voidaan poistaa rivejä. Tässä projektissa tietokantamalleja on luotu gallerioille, gallerioiden sisältä-

mille kuville sekä tekstialueille. Kuvassa 1 havainnollistetaan Django-sovelluksen visuaalista käyttöliittymää. Hallintasivulle määritellään osoite /hallinta projektin tiedostossa urls.py, mikä mahdollistaa helpon navigoinnin hallintasivulle.

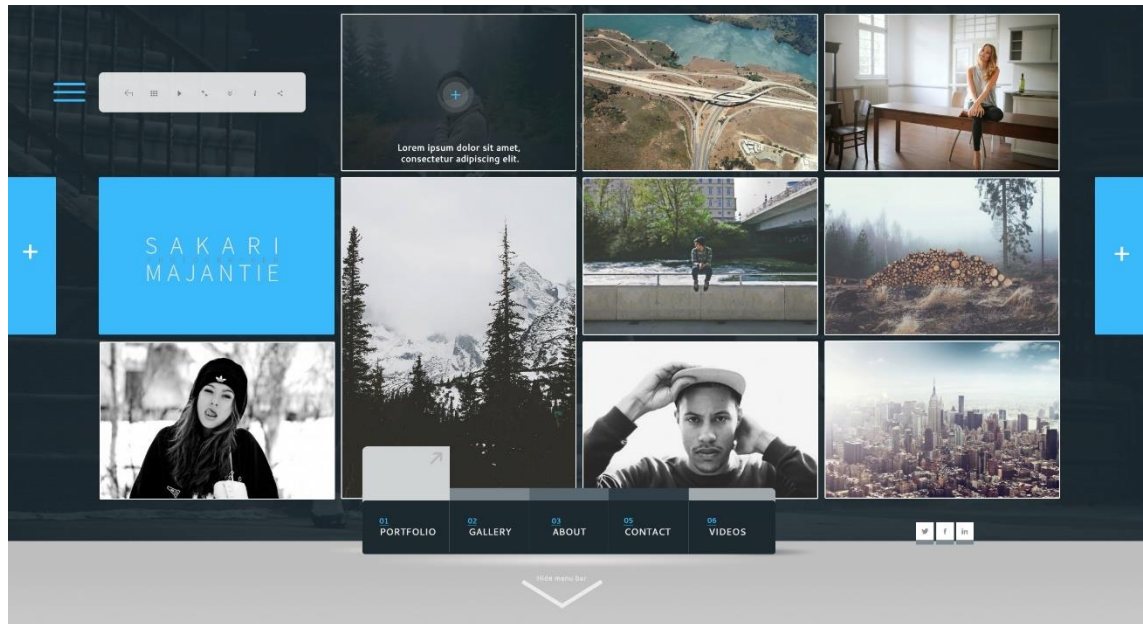
Sivuston ylläpito

Kirjautuminen ja oikeudet	
Käyttäjät	+
Ryhmät	+
Website	
Galleriat	+

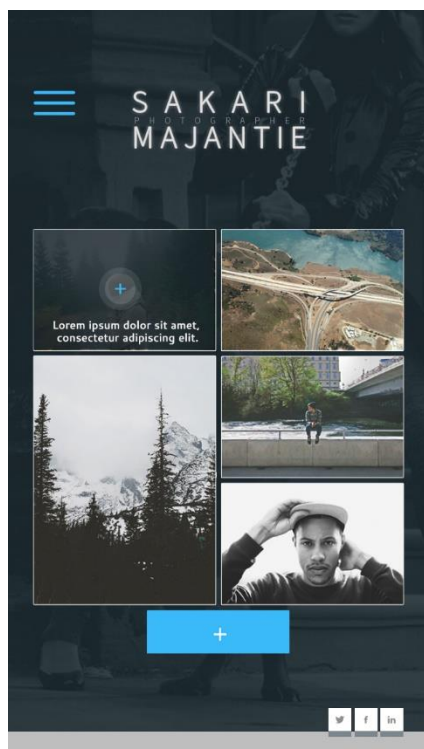
Kuva 1. Django-sovelluksen visuaalinen hallintakäyttöliittymä.

3.3 Ulkoasu

Alkuperäinen visuaalinen suunnitelma sivustoon on toimeksiantajan laatima. Tämä määrittelee suurimmilta osin sen, miten sivuston front end tulee rakentaa. Visuaalinen suunnitelma on tehty sekä työpöytä- että mobiilinäkymistä. Sivuston front end -puolen pohjalla käytetään Foundation 5 -ohjelmistokehystä, jonka päälle luodaan omia tyylejä sekä muokataan ohjelmistokehyksen mukana tulevia tyylejä. Foundation 5 -ohjelmistokehyksen ansiosta myös responsiivisuutta tuetaan projektissa alusta lähtien. Kuva 2 havainnollistaa sivustoa varten suunniteltua visuaalista suunnitelmaa työpöytänäkymässä. Ennen projektin alkua toimeksiantaja on toimittanut jokaiselta sivuston alisivulta vastaavan suunnitelman. Kuva 3 havainnollistaa mobiilikäyttöä varten suunniteltua visuaalista suunnitelmaa samasta alisivusta.



Kuva 2. Toimeksiantajan antama visuaalinen suunnitelma sivustoa varten työpöytänäköymässä.

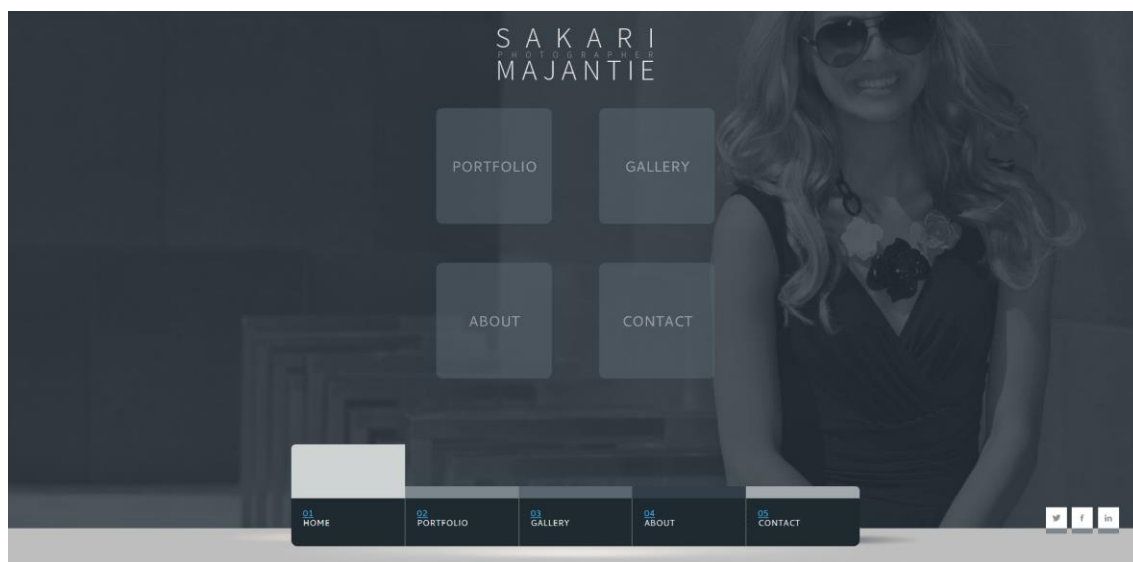


Kuva 3. Toimeksiantajan toimittama visuaalinen suunnitelma sivuston mobiilisivusta.

3.4 Osien toteutus

Rakenteellisia eroja tarkasteltaessa sivusto voidaan jakaa sisällöttömään etusivuun, galleriasivuihin sekä sivuihin, joissa on pääasiassa tekstisisältöä. Valikot ovat sivustolla globaaleja, eli ne ovat läsnä joka sivulla. Mobiilinäkymässä alhaalla oleva valikko katoaa näkyvistä, jolloin käyttäjän käytössä on yksi valikko. Muut sivuston toiminnallisuudet ovat sekä työpöytänäkymässä että mobiilinäkymässä pääosin samat, vaikkakin sivusto on tarkoituksella suunniteltu interaktiivisemmaksi työpöytäversiossa.

Ensimmäinen tavoite sivuston rakenteen luomisessa on saada aikaan pohja, johon muut sivuston alisivut voidaan perustaa. Käytettäessä samaa pohjaa sivuston kaikilla sivuilla saadaan sivustolle yhtenäinen visuaalinen ilme. Pohjan rakenne toteutetaan käyttäen elementtejä Foundation 5 -ohjelmistokehyksestä. Kaikilla alisivuilla läsnä olevan alapalkin valikon toimintaa säätelemään luodaan JQuerya hyödyntävä Javascript-tiedosto. Pohjasivun tarkoituksena on myös toimia sivuston pääsivuna sekä porttina, jonka kautta sivuston käyttäjät voivat navigoida haluamalleen alisivulle. Kuvassa 4 näkyy sivustolta otettu ruutukaappaus pohjasivulta ja kuvassa 5 näkyy vastaavan pohjasivun mobiilinäkymä.



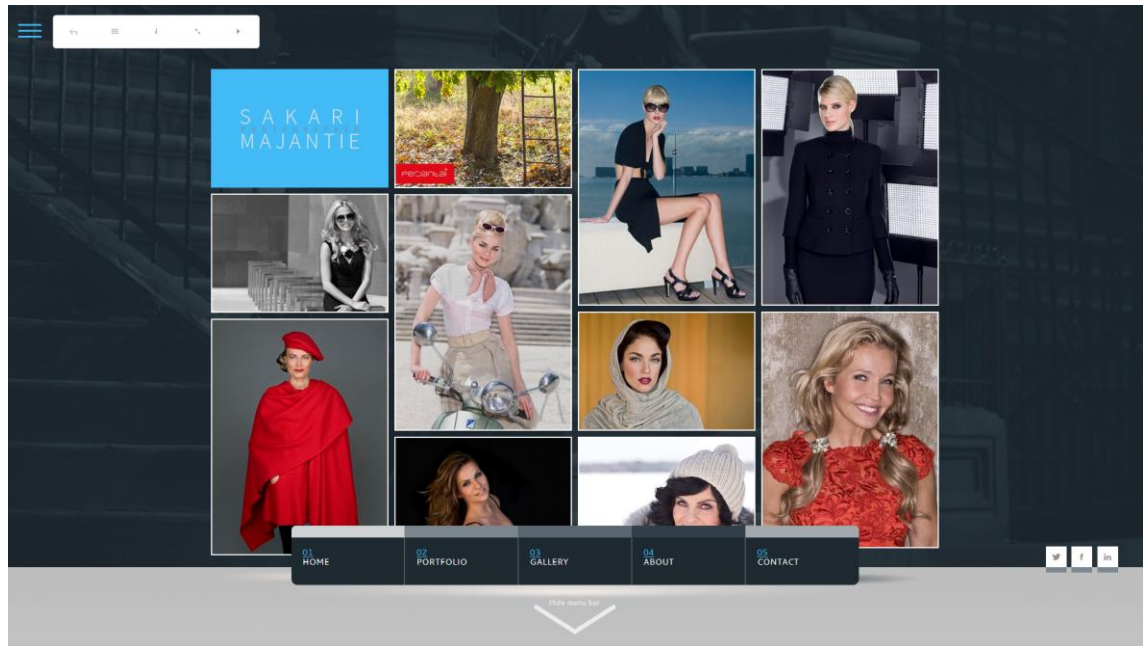
Kuva 4. Ruutukaappaus sivuston pääsivulta työpöytänäkymässä.



Kuva 5. Ruudunkaappaus sivuston pääsivulta mobiilinäkymässä.

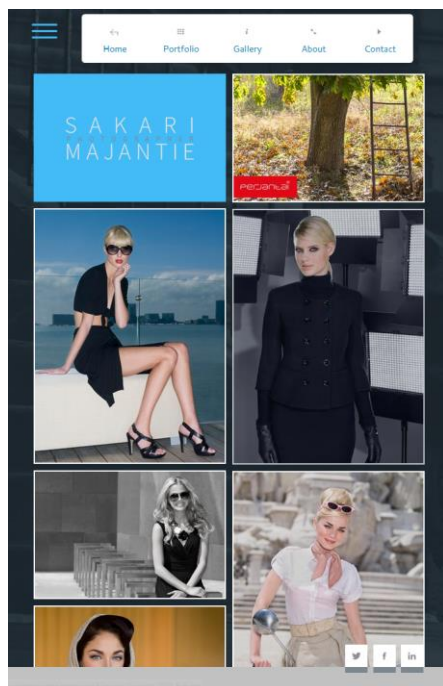
Toinen tavoite sivuston toteuttamisessa on luoda sekä työpöytäkäyttäjille että mobiilikäyttäjille toimiva ja selkeä galleriakäyttöliittymä. Kuvat on saatava uudelleenjärjestettyä responsiivisesti selainikkunan koon muuttuessa esimerkiksi tablettikäyttäjän kääntäessä ruutua. Responsiivinen galleriakäyttöliittymä mahdollistaa myös sen, että kaikille laitteille tarvitaan vain yksi galleriaelementti.

Halutun tuloksen aikaansaamiseksi käytetään Wookmark-Javascript-lisäosaa. Kuvassa 6 havainnollistetaan galleriasivua, jossa Wookmarkia on käytetty valokuvien järjestämiseen.



Kuva 6. Ruudunkaappaus sivuston galleriasivulta työpöytänäkymässä.

Kuva 7 havainnollistaa, miten Wookmark uudelleenjärjestää kuvat ruutukoon pienentyessä.



Kuva 7. Ruudunkaappaus galleriasivusta mobiilinäkymässä.

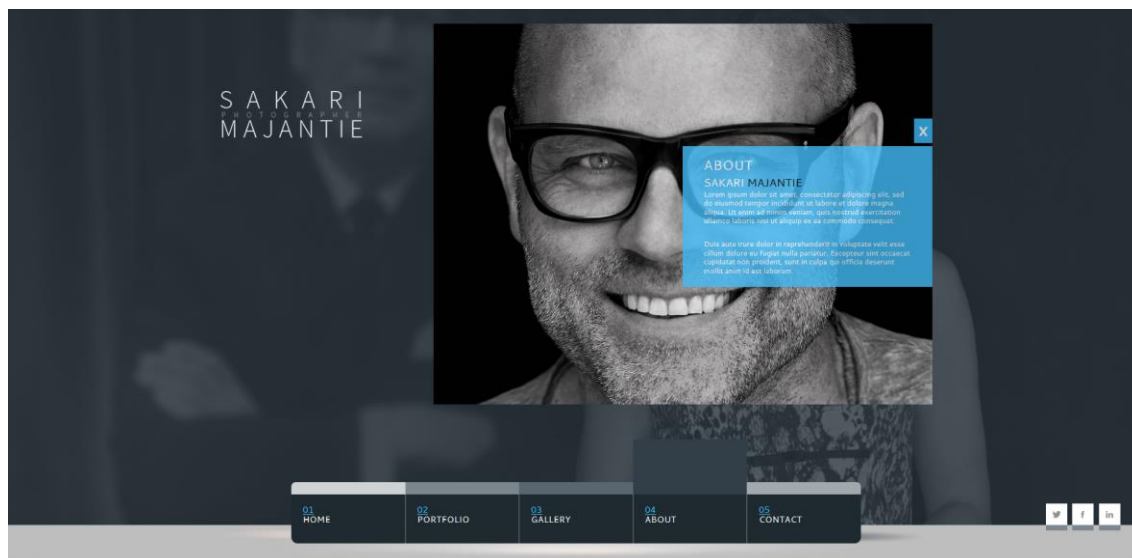
Koska kyseessä on valokuvaajan portfolioikäyttöön tuleva valokuvastudio, täytyy valokuvat saada sivustolle visuaalisesti näyttävällä tavalla. Suurin osa näyttävyydestä saadaan aikaan pelkästään toimeksiantajan visuaalista suunnitelmaa apuna käyttäen.

Kuvassa 8 näkyy, kuinka Lightbox-Javascript-lisäosa avaa klikatun kuvan suureksi. Lightbox käyttäytyy samoin myös mobiililaitteilla.



Kuva 8. Ruudunkaappaus tilanteesta, jossa sivuston galleriasivulla olevaa kuvaa on klikattu työpöytäkäytössä.

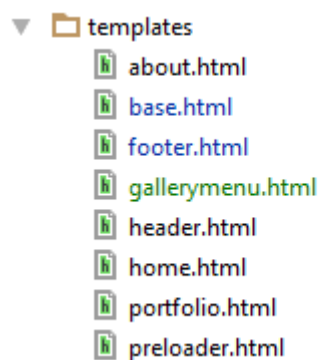
Kuvassa 9 näkyy tekstisisältösivu, jonka tarkoituksena on pääasiassa näyttää tekstiä. Tekstisisältö toimii vastapainona valokuville, joita sivustolla esiintyy paljon. Tekstisisällön tarkoituksena on ilmaista sivuston käyttäjille ne asiat, joita pelkät kuvat eivät voi ilmaista. Sivuston sisältösivuja ovat valokuvaajan tiedot -sivu sekä yhteydenottolomakesivu.



Kuva 9. Ruudunkaappaus sisältösivusta, jonka päätavoite on näyttää sivuston käyttäjille tekstiä.

3.5 Rakenteen toteutus

Django käyttää omaa template-kieltään. HTML:llä, Sassilla ja Javascriptillä luotu pohjasivu integroidaan Django-projektiin template-kielen avulla. Kuva 10 havainnollistaa, kuinka sivusto on jaettu HTML-tiedostoihin.



Kuva 10. Projektin template-tiedostokansion sisältö. Tekstin sininen ja vihreä väritys liittyvät versionhallintaan.

Template-kielen avulla koko sivuston rakenne koostetaan tiedostoon base.html. Koodi 4 havainnollistaa base.html-tiedoston body-tagin sisältöä.

```
<body>
```

```

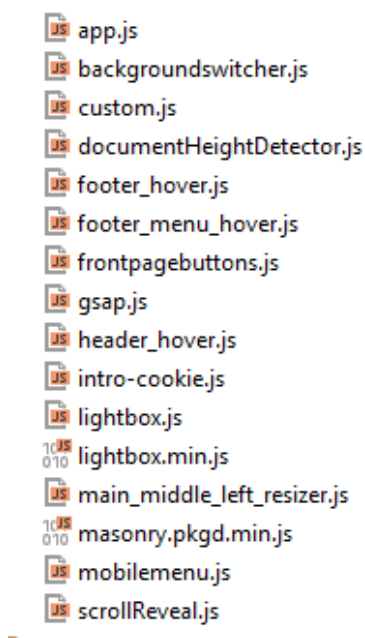
{% include 'header.html' %}
{% block content %}{% endblock %}
{% include "footer.html" %}
<div class="loaderdisplay">
{% include "preloader.html" %}
</div>
{% include "gallerymenu.html" %}

</body>

```

Koodi 4. base.html-tiedoston body-tagin sisältämä koodi, joka on sekoitus HTML- ja Django-templat-koodia.

Lisäksi yksinomaan projektin sivustoa varten kirjoitetaan useita Javascript-tiedostoja. Kuvassa 11 on kuvankaappaus projektin Javascript-tiedostokansion sisällöstä.



Kuva 11. Projektin Javascript-tiedostokansion sisältö.

Kaikki Javascript-kansion tiedostot eivät ole projektin nykyisessä vaiheessa käytössä. Merkittävimpiä projektia varten kirjoitettuja tiedostoja ovat tiedostot custom.js sekä footer_menu_hover.js, jotka sisältävät suurimman osan sivuston front end -puolen itsetehdystä toiminnallisuudesta.

SIVUSTON JULKAISU

3.6 Virtuaalipalvelimen hankinta

Sivuston isännöintiin (eng. hosting) käytetään virtuaalipalvelinta. Projektin sivustoa isännöivä virtuaalipalvelin hankitaan yritykseltä DigitalOcean. Palvelinta perustettaessa valitaan käyttöjärjestelmä, joka virtuaalipalvelin käyttää. Projektiin valittava palvelin käyttää 64-bittistä Ubuntu 12.04.5 -käyttöjärjestelmää. Lisäksi palvelinta luodessa valitaan sen sijainti sekä nimi. Palvelimen luomisen jälkeen annetaan palvelimen luojalle sen IP-osoite sekä salasana, jolla palvelimelle pääsee kirjautumaan sen "root"- eli hallintakäyttäjänä.

3.7 Virtuaalipalvelimen konfigurointi ja testaus

Konfiguroinnin eli asetusten asettamiseksi palvelimelle otetaan yhteys. Yhteyden muodostamiseksi käytetään PuTTY-ohjelmaa. Yhteys palvelimelle otetaan SSH-protokollaa käyttäen palveluntarjoajalta saaduilla tunnuksilla.

Ensin palvelimen asetuksiin määritellään käyttäjäryhmä ja tähän lisätään käyttäjä. Käyttäjäryhmälle annetaan kaikki palvelimen käyttöoikeudet. Tämän jälkeen poistetaan sisäänkirjautumisoikeudet root-käyttäjätunnukselta sekä lisätään sisäänkirjautumisoikeus uudelle käyttäjälle. Tämän toimenpiteen tarkoituksena on lisätä tietoturvaa palvelimelle lisäämällä ylimääräinen kerros turvajärjestelyjä vaatimalla käyttäjiltä yksi käyttäjätunnus ja salasana lisää ennen root-tunnuksia. [21]

Käyttäjienhallinnan jälkeen palvelimelle asennetaan Nginx-ohjelma, joka syöttää www-sivut www-sovellusta käyttävän sivuston selainikkunalle, sekä MySQL joka toimii projektin tietokantapalvelimena. Lisäksi palvelimelle asennetaan versionhallintaohjelma Subversion, jonka avulla palvelimella sijaitsevaa projektia voidaan päivittää suoraan versionhallinnan kautta. Tärkeää on myös asentaa palvelimelle Python ja Django sekä samanlainen virtuaaliympäristö kuin alkuperäisessä, paikallisessa projektissa on käytössä. Asennusten jälkeen palvelimelle ladattiin Subversionista projektin viimeisin versio, ja palvelin käynnistetään. [22]

Toistaiseksi palvelimeen on tehty ohjaus verkkotunnukseen *sakarimajantie.materiaali.fi*. Verkkotunnus ei ole sivustolle lopullinen, mutta sopii työn esittelyyn sekä testaukseen paremmin kuin pelkkä palveluntarjoajan antama IP-osoite palvelimelle.

Sivuston julkaisujärjestelmän testausta varten on navigoitava www-selaimella sivulle *sakarimajantie.materiaali.fi/hallinta*. Sivustolla oleva julkaisujärjestelmä käyttää alkuperäiseen verrattuna uutta MySQL-tietokantaa, joten mahdolliset paikallisessa projektissa sivustolle ladatut materiaalit on ladattava uudelle, internetissä sijaitsevalle sivustolle. Laddattaessa kuvia julkiselle sivustolle voidaan todeta, että julkaisujärjestelmä toimii tarkoitettulla tavalla.

4 LOPPUKÄYTTÄJÄN OHJEISTUS JA KOULUTUS

Loppukäyttäjän ohjeistusta varten laaditaan tekstidokumentti, jossa kuvaillaan, kuinka sivuston sisällön päivittäminen tapahtuu. Dokumentin sisältö rajataan ainoastaan niiden toimintojen kuvailuun, jotka loppukäyttäjän muokattavaksi on tarkoitettu (Liite 1).

5 TULOKSET JA JOHTOPÄÄTÖKSET

Työn tuloksena saatu www-sivusto täyttää toimeksiantajan vaatimukset. Työtä on helpottanut huomattavasti se, että sivuston visuaalinen suunnitelma on jo valmiina. Vaikein osuus pohjasivun tekemisessä on määritellä, miten alapalkin valikko toimii. Lisäksi työtä helpottaa se, että Django, jonka pohjalle projekti perustetaan, on laajalti toimeksiantajan käyttämä ja apua on tarvittaessa saatavilla toimeksiantajalta.

Oikealla työkalujen valinnalla on saatu aikaan haluttu tulos ilman ongelmatilanteita. Internetistä on mahdollista löytää laajalti informaatiota projektissa käytetyistä työkaluista. Lisäksi toimeksiantaja käyttää samoja työkaluja kuin projektissa käytetään, joten toimeksiantaja ymmärtää projektin rakenteen ja omaa tarvittavan tietotaidon projektin myöhempiä ylläpitoa varten. Vaihtoehtoiset työkalut projektin toteuttamiseksi olisivat olleet mahdollisia, mutta valitut työkalut ovat tässä tapauksessa osoittautuneet tarkoitukseen sopiviksi.

LÄHTEET

- [1] A List Apart, [www-dokumentti]. Saatavilla <http://alistapart.com/article/responsive-web-design> (luettu 6.6.2016).
- [2] Moveable Online, [www-dokumentti]. Saatavilla <http://moveableonline.com/blog/2013/10/29/7-advantages-using-cms-run-site/> (luettu 6.6.2016).
- [3] About Tech, [www-dokumentti]. Saatavilla <http://websearch.about.com/od/h/g/html.htm> (luettu 6.6.2016).
- [4] W3Schools, [www-dokumentti]. Saatavilla http://www.w3schools.com/css/css_intro.asp (luettu 6.6.2016).
- [5] JavaScripter, [www-dokumentti]. Saatavilla <http://www.javascripter.net/faq/> (luettu 6.6.2016).
- [6] Github, [www-dokumentti]. Saatavilla <https://github.com/germanysbestkeptsecret/Wookmark-jQuery> (luettu 6.6.2016).
- [7] Lokesh Dhakar, [www-dokumentti]. Saatavilla <http://lokeshdhakar.com/projects/light-box2/> (luettu 6.6.2016).
- [8] Github, [www-dokumentti]. Saatavilla <https://github.com/jlmakes/scrollreveal.js> (luettu 6.6.2016).
- [9] TechTerms, [www-dokumentti]. Saatavilla <http://techterms.com/definition/framework> (luettu 6.6.2016).
- [10] Zurb Foundation, [www-dokumentti]. Saatavilla <http://foundation.zurb.com/sites/docs/v/5.5.3/sass.html> (luettu 6.6.2016).
- [11] Python Software Foundation, [www-dokumentti]. Saatavilla <https://docs.python.org/3/faq/general.html> (luettu 6.6.2016).
- [12] Django Software Foundation, [www-dokumentti]. Saatavilla <https://docs.djangoproject.com/en/1.9/faq/general/> (luettu 6.6.2016).
- [13] Django Software Foundation, [www-dokumentti]. Saatavilla <https://docs.djangoproject.com/en/1.9/howto/deployment/wsgi/> (luettu 6.6.2016).

- [14] Django Software Foundation, [www-dokumentti]. Saatavilla <https://docs.djangoproject.com/en/1.9/intro/overview/> (luettu 6.6.2016).
- [15] Django Software Foundation, [www-dokumentti]. Saatavilla <https://docs.djangoproject.com/en/1.9/topics/install/#installing-official-release> (luettu 6.6.2016).
- [16] Django Software Foundation, [www-dokumentti]. Saatavilla <https://docs.djangoproject.com/en/1.9/ref/contrib/admin/> (luettu 6.6.2016).
- [17] Django Software Foundation, [www-dokumentti]. Saatavilla <https://docs.djangoproject.com/en/1.9/intro/tutorial01/> (luettu 6.6.2016).
- [18] The Hitchhiker's Guide to Python, [www-dokumentti]. Saatavilla <http://docs.python-guide.org/en/latest/dev/virtualenvs/> (luettu 6.6.2016).
- [19] Django Software Foundation, [www-dokumentti]. Saatavilla <https://docs.djangoproject.com/en/1.8/ref/settings/#databases> (luettu 6.6.2016).
- [20] Django Software Foundation, [www-dokumentti]. Saatavilla <https://docs.djangoproject.com/en/1.9/intro/tutorial02/> (luettu 6.6.2016).
- [21] Media Temple, [www-dokumentti]. Saatavilla <https://mediatemple.net/community/products/dv/204643810/how-do-i-disable-ssh-login-for-the-root-user> (luettu 6.6.2016).
- [22] Digital Ocean, [www-dokumentti]. Saatavilla <https://www.digitalocean.com/community/questions/what-is-nginx-and-why-might-i-want-to-use-it-over-apache> (luettu 6.6.2016).

Aleksi Korpelainen

aleksi.korpelainen@perjantai.fi

digi@perjantai.fi

sakarimajantie.materiaali.fi-ohje sivustonhallintaan

Hallintapaneeliin kirjautuminen

Navigoi selaimella osoitteeseen sakarimajantie.materiaali.fi/hallinta. Kirjaudu sisään saamillasi tunnuksilla. Kirjaututtuasi sisään sinun tulisi nähdä teksti "Site administration" ja tämän tekstin alla linkkejä sivustonhallinnan osioihin. Tämä on hallintapaneelin etusivu.

Käyttäjien hallinta

Hallintapaneelin etusivulla klikkaa kohtaa "Users". Täällä näet kaikki sivustonhallintaan luodut käyttäjät. Muokataksesi käyttäjän tietoja tai oikeuksia klikkaa käyttäjätunnuksen nimeä. Aukeavassa valikossa voit muokata käyttäjätunnusta, salasanaa, henkilötietoja sekä oikeuksia. Tällä sivustolla ainoastaan "Superuser status" on merkitsevä oikeus, ja jos se poistetaan käyttäjältä, hän ei näe hallintavalikossa mitään. Tehtyäsi tarvittavat muutokset klikkaa oikeassa alakulmassa olevaa "Save"-linkkiä.

Luodaksesi käyttäjän klikkaa käyttäjienhallinnan oikeassa yläkulmassa näkyvää "Add user"-linkkiä. Tästä aukeaa ikkuna, jossa pyydetään ensin käyttäjänimi sekä salasana uudelle käyttäjälle. Kun nämä on tehty, ilmestyy uusi käyttäjä käyttäjienhallintaan.

Valokuvien hallinta

Klikkaa hallintapaneelin etusivulla kohtaa "Galleriat". Tästä aukeava galleriavalikko sisältää kaikki sivustolla näkyvät kuvagalleriat. Gallerioiden nimiin ei tulisi koskea, sillä galleriat ilmestyvät sivuille nimen perusteella. Jos kuitenkin vahingossa poistat gallerian, voit luoda uuden gallerian samalla nimellä, jolloin kuvat näkyvät taas sivustolla.

Muokataksesi gallerian kuvia klikkaa jokin galleria näkyviin. Tässä näkymässä voit lisätä tai poistaa gallerian kuvia. "Browse..."-painikkeista voit ladata kuvia galleriaan tietokoneeltasi. Painikkeen vieressä oleva tekstikenttä määrittelee tekstin, joka kuvan yhteydessä näytetään. Tämän voi halutessaan jättää tyhjäksi. Tämän tekstikentän oikealla puolella on valintalaatikko, jolla määritellään sivustolla näkyvän kuvan pienoysversion asettelu. Mikäli kuva on pystymallinen, kannattaa valintalaatikkoon klikata rasti. Tällöin

myös pienoiskuvasta tulee pystymallinen. Aivan oikealta löytyvät nuolen sekä rastin kuvat. Nuolesta voi hiirtä pohjassa pitämällä ja vetämällä vaihtaa kuvien järjestystä. Rastista tapahtuu kuvan poisto. Kun olet tyytyväinen gallerian muutoksiin, voit tallentaa gallerian oikeassa alakulmassa olevalla tallennuspainikkeella.

Lomakevastauksien katselu ja poisto

Klikkaa hallintapaneelin etusivulla linkkiä "Lomakevastaukset". Tähän kerätään kaikki sivuston palautelomakkeeseen kirjoitetut vastaukset. Yksityiskohtaisesti vastausta pääsee tarkastelemaan klikkaamalla vastauksen lähetyspäivämäärässä olevaa linkkiä. Mikäli listassa esiintyy epämääräisiä vastauksia (esimerkiksi omituisia linkkejä, joita ei yleensä suositella klikkaamaan), voidaan ne poistaa klikkaamalla poistettavien vastaus-ten vieressä olevaa valintalaatikkoa ja tämän jälkeen valitsemalla vasemmassa alakulmassa olevasta alasvetovalikosta kohta "Delete". Tällöin sivusto kysyy sinulta varmistusta poistoon, jonka voit kuitata painamalla oikealla olevaa "Yes, I'm sure" -linkkiä.