



**LAHDEN AMMATTIKORKEAKOULU**  
*Lahti University of Applied Sciences*

# SIVUSTON TOTEUTUS DRUPALILLA

LAHDEN  
AMMATTIKORKEAKOULU  
Tekniikan ala  
Mediatekniikan koulutusohjelma  
Tekninen visualisointi  
Opinnäytetyö  
Kevät 2014  
Markus Kangas

Lahden ammattikorkeakoulu  
Mediatekniikka

KANGAS, MARKUS:

Sivuston toteutus Drupalilla

Suuntautumisvaihtoehdon opinnäytetyö, 38 sivua

Kevät 2014

TIIVISTELMÄ

---

Tämä opinnäytetyö käsittelee www-sisällönhallintajärjestelmiä, joista erityisesti Drupalia. Drupalista käydään läpi perusteita ja case-osudessa Marioff.com-sivuston referenssigallerian teko.

Drupal on tullut tutuksi aluksi omien harrastusten kautta ja sen jälkeen työelämässä. Case-osuudessa otetaan teoriaosaaminen käyttöön ja suunnitellaan sekä toteutetaan toimiva referenssigalleria käyttäen apuna Drupalin omia toimintoja.

Asiasanat: Drupal, sisällönhallintajärjestelmä, cms, php, koodaaminen

Lahti University of Applied Sciences  
Degree Programme in Media Technology

KANGAS, MARKUS:

Site implementation with Drupal

Bachelor's Thesis in visualization engineering, 38 pages

Spring 2014

ABSTRACT

---

This thesis deals with web content management systems including Drupal in particular. go through basics of Drupal and in the case section making of Mari-off.com sites reference gallery.

Drupal has become familiar at first with own hobbies and thereafter in employment. In the case section purpose is to take use of the theory of knowledge and to design and implement a reference gallery using Drupals own systems.

Key words: Drupal, content management system, cms, php, coding

## SISÄLLYS

1	JOHDANTO	1
2	SISÄLLÖNHALLINTAJÄRJESTELMISTÄ	2
3	DRUPAL	4
3.1	HTTP-pyyntöjen käsittely Drupalissa	4
3.2	Tiedon kulku	6
3.3	Järjestelmävaatimukset	7
3.4	Arkkitehtuuri	8
3.4.1	Nodet eli solmut	8
3.4.2	Fieldit eli kentät	8
3.4.3	Blokit ja regionit	9
3.4.4	Hookit	10
3.4.5	Taksonomia	11
3.4.6	Käyttäjät, roolit ja käyttöoikeudet	11
4	CASE: MARIOFF.COM-SIVUSTON TOTEUTUS	13
4.1	Drupalin asennus	13
4.2	Moduulien asennus	16
4.3	Sisältötyyppien luonti	17
4.4	Taxonomiat	18
4.5	Moduulin luonti	19
4.6	Blokit	24
4.7	Teeman luonti	30
4.7.1	.info-tiedosto	30
4.7.2	Template.php	32
5	YHTEENVETO	35
	LÄHTEET	36

## 1 JOHDANTO

Tässä opinnäytetyössä käydään läpi yleisellä tasolla sisällönhallintajärjestelmät ja keskitytään tarkemmin Drupaliin, jolla tämän työn case-osuuskin on toteutettu.

Drupalia pidetään aluksi vaikeana omaksua, ja tämän opinnäytetyön tarkoitus on avata hieman Drupalin perusteita ja case-osuuden esimerkkien avulla käydä niitä läpi. Työssä ei keskitytä ohjelmoinnin perusteisiin ja oletuksena on, että perusteet niin PHP:stä, HTML:stä ja CSS:stä ovat hallussa, koska Drupalin kehitys niitä vaatii.

Opinnäytetyön case on osa Marioff.com-sivusta. Sivusto toteutettiin Marioff Corporation Oy:lle ja toteuttamisesta vastasi Verkkojulkaisut Oy. Sivusto kokonaisuudessaan olisi liian laaja käsitellä kokonaan läpi tässä opinnäytetyössä, joten tähän valittiin referenssi galleriantoteutus, missä käydään läpi useita osa-alueita Drupalista.

Verkkojulkaisut on Helsingissä toimiva digitoimisto, jonka toimialueisiin kuuluu verkkosivujen ja -palveluiden rakentamisen, ylläpidon ja kehittämisen lisäksi myös sosiaalisen median ohjelmistot, hakukoneoptimoinnit, kampanjat sekä sähköinen suoramarkkinointi (Verkkojulkaisut, 2014).

Tärkeimpänä yrityksen työvälineenä verkkosivustoja rakentaessa on tällä hetkellä Drupal. Drupalin suosio perustuu avoimeen lähdekoodiin ja maksuttomuuteen sekä maailmanlaajuiseen yhteisöön, joka vastaa Drupalin jatkokehityksestä. Asiakkaatkin voivat näin ollen hyödyntää ja jatkokehittää avoimeen lähdekoodiin perustuvia ratkaisuja (Verkkojulkaisut, 2014).

## 2 SISÄLLÖNHALLINTAJÄRJESTELMISTÄ

Sisällönhallintajärjestelmä eli CMS (engl. Content Management System) on tietokonejärjestelmä, jolla loppukäyttäjä voi lisätä sekä muokata sisältöä yhdestä keskitetystä käyttöliittymästä (Wikipedia 2014c).

Tämä työ käsittelee tarkemmin www-sisällönhallintajärjestelmiä, mutta muitakin sisällönhallintajärjestelmiä on olemassa, kuten

- dokumenttienhallintajärjestelmä
- julkaisujärjestelmä (yleisnimitys julkaisupainotteiselle sisällönhallintajärjestelmälle, mutta myös usein synonyymi www-sisällönhallintajärjestelmälle tai www-julkaisujärjestelmälle)
- verkkokauppajärjestelmä
- aineistonhallintajärjestelmä (muun muassa kuva-aineistot, videomateriaali, multimedia)
- Wiki-järjestelmät

(Wikipedia 2014c).

Yksi www-sisällönhallintajärjestelmän eduista on halpuus, koska suurimman osan voi käydä suoraan lataamassa kyseisen järjestelmän omilta sivuilta. Helppokäyttöisyys on myös yksi eduista, koska sisältöä voi hallita ilman minkäänlaista tietotaitoa ohjelmoinnista. Sivuston muokattavuus helpottuu myös, koska yhteisön tekemiä lisäosia ja teemoja löytyy valmiina valtavasti. Järjestelmät ovat usein myös valmiiksi hakukoneystävällisiä (Wikipedia 2014a).

Yleisimpiä www-sisällönhallintajärjestelmiä tälläheikellä ovat Wordpress, Joomla ja Drupal (W3techs 2014). Kaikki ovat ilmaisia avoimeen lähdekoodiin perustuvia sisällönhallintajärjestelmiä.

Wordpress on tällä hetkellä ylivoimaisesti suosituin www-sisällönhallintajärjestelmä johtuen helppokäyttöisyydestä ja valmiiden lisäosien määrästä, joita on

enemmän kuin millään muulla. Yli 60 % www-sisällönhallintajärjestelmistä internetissä käyttää Wordpressiä alustanaan. (Websitesetup 2014.)

Drupal on sisällönhallintajärjestelmän lisäksi myös sisällönhallintarajapinta eli CMF (engl. Content management framework), jolla voi tehdä omia sisällönhallintajärjestelmiä. Tämä onkin yksi Drupalin vahvuuksista, koska sen voi räätälöidä omiin tai asiakkaan tarpeisiin paljon paremmin. Drupal ei ole yhtä helppokäyttöinen kuin Wordpress tai Joomla, johtuen suuresta muokkattavuudesta. Onkin erittäin suositeltavaa, että käyttäjä hallitsee edes jonkin verran PHP:tä, HTML:ää, CSS:ää ja javascriptiä saadakseen Drupalista enemmän irti.

Joomla sijoitetaan vaikeudeltaan yleisesti Wordpressin ja Drupalin välimaastoon (Websitesetup 2014). Se ei vaadi yhtä paljon teknistä osaamista kuin Drupal, ja siinä on enemmän ominaisuuksia kuin Wordpressissä, kuitenkin säilyttäen yksinkertaisuuden ja helppokäyttöisyyden.

### 3 DRUPAL

Drupal on kirjoitettu PHP-ohjelmointikielellä, ja sitä jaetaan GNU-hankkeen yleisen lisenssin alaisuudessa (engl. GNU Genereal Public Licence GPL) (Drupal 2014e). GPL-lisenssi antaa kaikille oikeuden käyttää, muokata, muuttaa ja edelleen levittää ohjelmia sekä niiden lähdekoodeja (Wikipedia 2014b). Tämän ansiosta Drupaliin on aktiivisen yhteisön toimesta ohjelmoitu yli 25 000 lisäosamoduulia ja lähes 2 000 teemaa, jotka voi suoraan ladata Drupalin omilta sivuilta (Drupal 2014a).

Drupalin coreksi eli ytimeksi kutsutaan tiedostoja, jotka ovat mukana Drupalin sivuilta ladatussa paketissa (Hodhdon 2013, 18). Ydin sisältää

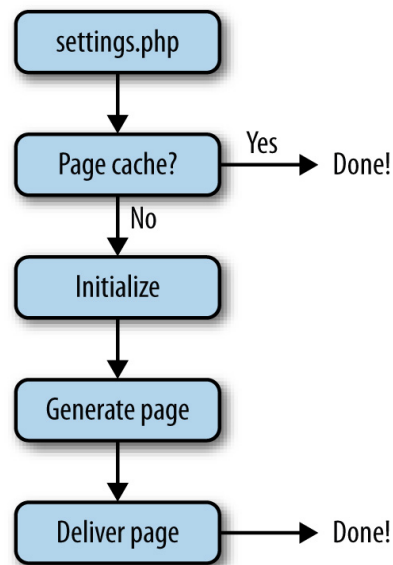
- joustavan käyttäjähallintajärjestelmän käyttäjäoikeuksien ja roolien säätämiseen
- sisällön muokkauksen suoraan sivulla
- teemajärjestelmän, jonka avulla voi muokata näkymiä napeista koko sivuun
- blokki-moduulin, joka mahdollistaa sisältöjen osien näyttämisen sivujen eri alueilla.
- valikkonavigoinnin
- taksonomia-moduulin sisällön kategorioimiseen ja ”täggaämiseen”
- graafisen järjestelmälokin virheille ja tapahtumille

(Hodhdon 2013, 18).

#### 3.1 HTTP-pyyntöjen käsittely Drupalissa

Kun palvelin saa http-pyyntön, joka vastaa Drupal-sivua, index.php-tiedosto ladataan ja suoritetaan. Drupal käsittelee http-pyyntön kuvan 1 kaavion mukaan.





KUVA 1. http-pyynnön käsittelykaavio (Hodhdon 2013, 5)

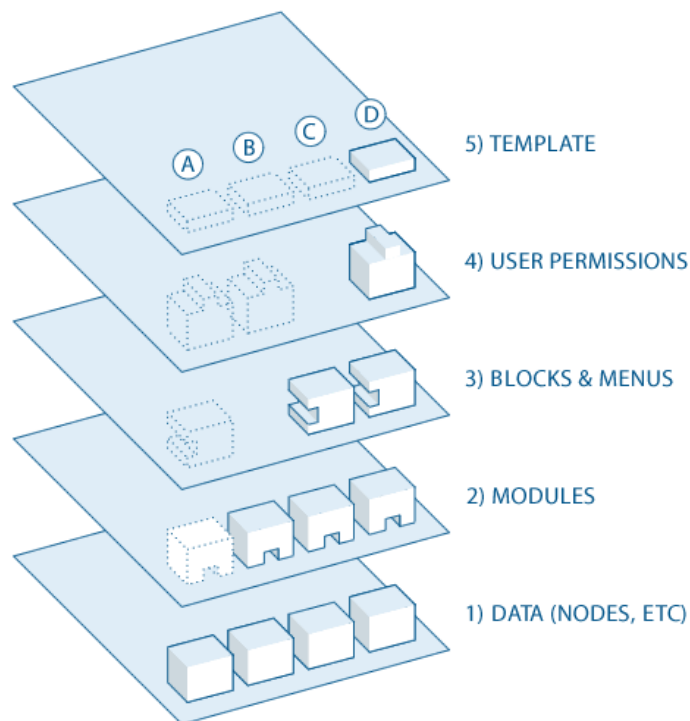
Drupal käsittelee http-pyynnot seuraavanlaisesti

1. Drupal päättelee, mitä settings.php:tä käytetään, mikä johtuu siitä, että Drupalissa voi olla useita sivuja eri settings.php:eillä. Tällöin kyseessä on niin sanottu Multisite.
2. Jos käyttäjä on anonymi, eli ei ole kirjautunut sisään, tarkistetaan, onko sivu tallennettu välimuistiin edellisellä http-pyynnöllä. Jos näin on, tämä sivu palautetaan käyttäjälle. Drupalin sivujen välimuistitus ei päde sisäänkirjautuneille käyttäjille.
3. Muodostetaan tietokantayhteys sekä PHP session -muuttujat.
4. Käynnistetään kielijärjestelmä ja ladataan ytimen vaatimat tiedostot sekä päälläolevien moduulien .module-tiedostot.
5. Tarkistetaan, onko sivu normaali- vai huoltotilassa.
6. Jos sivu on huoltotilassa, haetaan adminin asettama huoltotilaviesti ja näytetään se käyttäjälle.
7. Jos sivu on normaalitilassa tai sisäänkirjautunut käyttäjä hakee huoltotilassa olevaa sivua, Drupal päättelee, mitä funktioita pitää kutsua, jotta saadaan generoitua oikea sisältö käyttäjälle.

8. Drupal määrittelee tarvittavat funktiot tulostettavalle sivulle ja kutsuu niitä.
9. Sisällöstä muodostetaan HTML-dataa, joka tulostetaan käyttäjälle.  
(Hodhdon 2013, 20 – 21.)

### 3.2 Tiedon kulku

Drupalissa on viisi tasoa, joita pitkin informaatio kulkee (kuva 2).



KUVA 2. Drupal tasot

#### Drupalin tasot

1. Alimpana on sivuston data ja sisältö. Ennen kuin sivulla voidaan näyttää mitään, se täytyy olla tallennettuna tietokantaan.
2. Seuraavalla tasalla ovat moduulit eli lisäosat, joko Drupalin ytimen omat, yhteisön tuottamat moduulit tai itse koodatut moduulit.

3. Moduulien jälkeen tulevat blokit ja valikot. Blokit voivat näyttää mitä sisältöä tahansa, ja ne voidaan sijoittaa mille tahansa teeman ennalta määritetälle alueelle sivulla. Valikkoja käytetään navigoimiseen sekä loogisten polkujen luomiseen sisällöille
4. Tämän jälkeen on vuorossa käyttöoikeudet, joilla määritellään mitä kullekin käyttäjälle näytetään sivulla.
5. Viimeisellä tasolla on teema, joka koostuu lähinnä HTML:stä, CSS:stä ja PHP-muuttujista, jotka takaavat, että näytettävät asiat ovat oikeilla paikoillaan sivulla.

(Drupal 2014e.)

### 3.3 Järjestelmävaatimukset

Järjestelmävaatimukset pätevät vain Drupalin versioon 7.x, muita versioita ei ole huomioitu, koska ne eivät ole tämän työn kannalta oleellisia.

Minimiasennus vaatii vähimmillään 15 MB kiintolevytilaa palvelimelta, mutta ottaen huomioon lisämoduulien, teemojen ja kaikkien kuvien sekä videoiden vaatiman tilan, olisi kiintolevytilaa oltava reilusti enemmän (Drupal 2014d).

Www-palvelimeksi suositellaan vähintään Apachen versiota 2.x , koska suurin osa Drupal-kehityksestä ja testauksesta tapahtuu Apache-palvelimella. Drupal voidaan asentaa myös Nginx- sekä Microsoft IIS -palvelimille. (Drupal 2014d).

Tietokannaksi suositellaan MySQL versiota 5.0.15 ja uudemmat. Myös PostgreSQL 8.3 sekä SQLite 3.3.7 sekä uudemmat versiot ovat tuettuja (Drupal 2014d).

Php:n suositeltu versio on 5.3, ja minimissään sen tulisi olla versiota 5.2.5. Tämän lisäksi PDO-tietokantalaajennus tulee olla päällä. PHP:n vaatima muistiraja (memory\_limit) tulisi olla minimissään 32 MB, mutta yleensä käytössä on erilaisia moduuleita, jotka vaativat tuon määrän nostamista. (Drupal 2014d).

### 3.4 Arkkitehtuuri

Drupal on modulaarinen järjestelmä, joka on koottu monesta pienestä osasta. Koska Drupal on sisällönhallintajärjestelmän lisäksi myös sisällönhallintarajapinta, vaaditaan käyttäjältä tietotaitoa myös Drupalin arkkitehtuurista ja termistöstä, jotta siitä saisi enemmän irti.

#### 3.4.1 Nodet eli solmut

Drupalin kanssa tekemisissä oleva ei voi välttyä asialta nimeltä node, joka on yksi Drupalin rakenteen peruspilareista. Node on kaikessa yksinkertaisuudessaan joukko toisiinsa liittyviä tietoja (Drupal.org 2014e). Esimerkiksi node tyypiltään uutinen, jossa voi tekstikentän lisäksi olla otsikko, pääkuva, kirjoittaja ja luomisaika.

Kaikki sisältötyypit ovat nodeja, ja niitä on erittäin helppo muokata suoraan Drupalin käyttöliittymästä. Kenttiä voi lisätä, poistaa ja editoida. Myös se, miten kenttiä näytetään sivulla, voidaan hallita suoraan Drupalin käyttöliittymästä.

#### 3.4.2 Fieldit eli kentät

Sisältötyyppien erot tulevat paremmin esiin niihin lisättyjen kenttien ansiosta. Kentän tyyppi määrittelee, minkä muotoista dataa voidaan kenttään tallentaa. Se, miten kenttään tallennetaan tietoa, määritellään kentän widgetissä, kuten tekstikenttä, monivalintalista tai tiedoston lähetysoikeus (Kuva 3). (Drupal 2014b.)

LABEL	MACHINE NAME	FIELD TYPE	WIDGET	OPERATIONS
Title	title	Node module element		
Body	body	Long text and summary	Text area with a summary	edit delete
Add new field				
<input type="text"/>		Long text	Text area (multiple rows)	
Label		Type of data to store.	Form element to edit the data.	
Add existing field				
<input type="text"/>	- Select an existing field -		- Select a widget -	
Label	Field to share		Form element to edit the data.	

### KUVA 3. Kenttien hallinta

Jokaiselle kentälle voidaan myös määritellä näkyvyysasetukset, kun käyttäjälle näytetään tallennettua nodea. Kenttien otsikon voi poistaa näkyvistä ja tekstin formaattiin voi vaikuttaa tai kentän voi ottaa kokonaan pois näkyvistä. Kuvien näyttökoon voi myös valita ennaltamääritellyistä tai itsetehdyistä kuvatyyleistä. Tulostusjärjestykseen voi myös vaikuttaa painokertoimen avulla joko raahaamalla kentät oikeaan järjestykseen tai antamalla niille painoarvot, jos painoarvot on laitettu näkyviin.

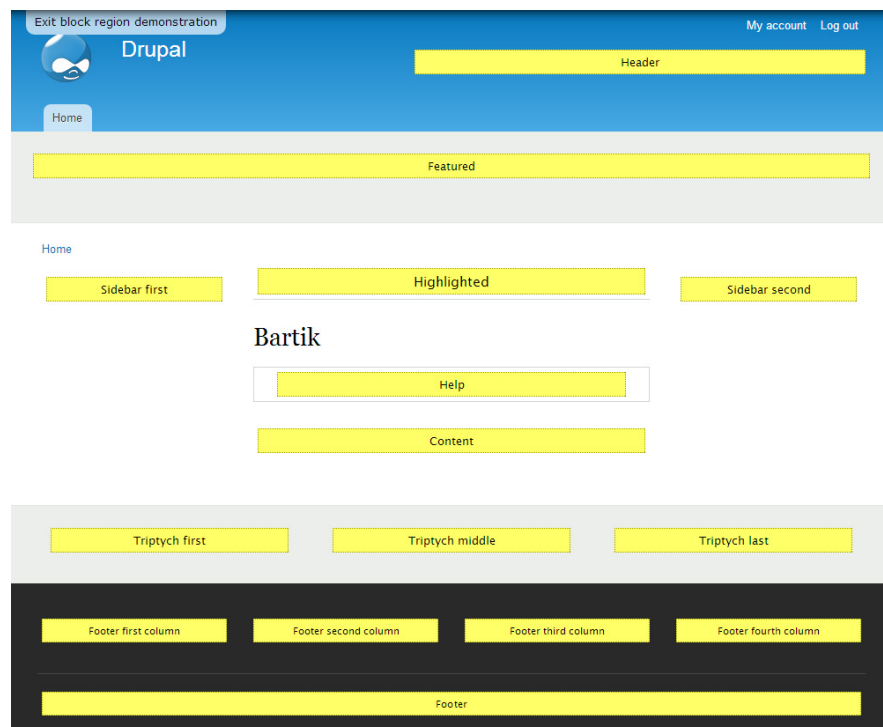
#### 3.4.3 Blokit ja regionit

Blokit ovat sisältölaatikoita joita voidaan sijoittaa sivun teeman eri alueille (Drupal 2014f). Blokkien näkyvyyteen vaikuttavat seuraavat asiat:

- Onko blokki laitettu sivulle, jota katsotaan?
- Onko sivulla kyseinen regioni, johon blokki on sijoitettu?
- Onko käyttäjällä tarvittavat käyttöoikeudet blokin näkemiseen?

(Shreves & Dunwoodie 2011, 48 - 49)

Regionit ovat ennaltamääriteltäviä alueita sivulla, johon blokkeja sekä sisältöjä voidaan sijoittaa (kuva 4). Regionit määritellään teeman .info-tiedostossa, josta lisää casen teeman luonnissa.



KUVA 4. Regionien demonstrointi

#### 3.4.4 Hookit

Koko Drupalin moduulisysteemi perustuu niin sanottuihin hookeihin, jotka ovat vuorovaikutuksessa Drupalin coren eli ytimen kanssa. Jotta Drupaliin voi kirjoittaa omia custom-moduuleita, on ymmärrettävä hieman hookien toimintaa. Lähes jokaisen sivun palasen rakentaminen sisältää useita hook-kutsuja, joilla voi vaikuttaa Drupalin toimintaan.

Hookit ovat PHP-funktioita, joiden nimi muodostuu rakenteella moduulin nimi\_hookin nimi(). Jokaisella hookilla on ennaltamääritellyt parametrit sekä paluuarvot.

Hookeja on niin paljon, että niihin kannattaa tutustua Drupalin sivuilla <http://api.drupal.org/api/drupal/includes!module.inc/group/hooks/7>, josta löytyvät myös tarpeelliset tiedot hookeista sekä käyttöesimerkkejä.

### 3.4.5 Taksonomia

Taksonomia on sisällönhallinnan organisointiin tarkoitettu ydin moduuli. Taksonomian avulla käyttäjä pystyy määrittelemään yksittäisiä termejä sekä sanastoja (engl. vocabulary), jotka ovat termien ryhmiä. Sanasto voi olla hierarkkinen, voi määritellä yksittäis- tai monivalinnan ja termejä voi myös lisäillä sanastoon automaattisesti uutta sisältöä luodessa tai muokattaessa. Jokainen sanasto voidaan liittää yhteen tai useaan sisältötyyppiin, tällä tavoin sisältöjä saadaan ryhmiteltyä kategorioihin. Suuri osa taksonomian käytöstä liittyy siihen, että saadaan samoilla termeillä olevat sisältötyypit listattua yhdessä helposti. Drupalin ydin tarjoaa tähän oletuksena url-polun taxonomy/term/8, missä 8 viittaa termin ID-arvoon. (Melançon ym. 2011, 222 - 224).

### 3.4.6 Käyttäjät, roolit ja käyttöoikeudet

Jokaista sivustolla vierailevaa pidetään Drupalissa käyttäjänä ja jokaiselle käyttäjälle voi antaa yhden tai useamman roolin johon on määritelty käyttöoikeudet (Melançon ym. 2011, 229).

Drupalin standardiasennukseen on mukana kolme roolia:

- Anonymous user: kaikki käyttäjät, jotka eivät ole kirjautuneet sisään.
- Authenticated user: sisään kirjautuneet käyttäjät.
- Administrator: rooli, joka saa automaattisesti kaikki käyttöoikeudet, kun uusi moduuli on otettu käyttöön.

NAME	OPERATIONS
 anonymous user (locked)	<a href="#">edit permissions</a>
 authenticated user (locked)	<a href="#">edit permissions</a>
 administrator	<a href="#">edit role</a> <a href="#">edit permissions</a>
<input type="text"/>	<input type="button" value="Add role"/>

KUVA 5. Admin-näkymän käyttäjäroolit

Kaksi ensimmäistä ovat oleellisia Drupalin toiminnan kannalta, joten niitä ei voi poistaa. Käyttäjäroolien määrää sivustolla ei ole rajoitettu ja jokaiselle roolille voidaan määritellä tietyt käyttöoikeudet, jolla kontrolloidaan sitä, mitä käyttäjä voi sivustolla tehdä (Melançon ym. 2011, 229 - 231).

Kaikki sisäänkirjautuneet käyttäjät perivät käyttöoikeudet authenticated user -roolilta, mutta ei mitään anonymous user -roolilta. Tämä siksi, koska kaikki roolit lukuunottamatta anonymous user -roolia vaativat sisäänkirjautumisen.



## 4 CASE: MARIOFF.COM-SIVUSTON TOTEUTUS

Marioff on maailman johtava vesisumusammutusteknologian kehittäjä, joka toimittaa palosammutusjärjestelmiä tuotenimellä HI-FOG®. HI-FOG® rajoittaa, tukahduttaa ja sammuttaa tulipaloja laukaisemalla korkealla paineella hienoa vesisumua palokohteeseen. Vesisumu katkaisee säteilylämmön ehkäisten palon leviämistä. Perinteiseen sprinklerijärjestelmään verrattuna HI-FOG® käyttää huomattavasti vähemmän vettä, joten sammutustyöstä aiheutuu minimaalisesti vesivahinkoja. Marioff on osa United Technologies konsernia ja työllistää maailmanlaajuisesti yli 400 henkeä.


Ennen sivuston varsinaista toteutusta käytiin projektipäällikön kanssa projekti läpi kaikkien teknisten ominaisuuksien sekä toimintojen osalta. Mitä sisältötyyppejä ja taksonomioita tulisi olemaan ja miten referenssigallerian ja yhteystietosivun tulisi toimia.

Tässä työssä käydään läpi tarkemmin Marioff-sivun referenssigallerian toteutus, koska koko sivuston läpikäyminen olisi liian laaja. Referenssigalleriassa käydään läpi sisältötyyppien luonti, taksonomiat, moduulin ohjelmointi ja sivujen teemaus.

### 4.1 Drupalin asennus

Ensiksi siirretään Drupalin tiedostot palvelimelle ja mennään selaimella kyseiseen osoitteeseen, jolloin Drupal tunnistaa automaattisesti, ettei tietokantaa ole, ja palauttaa käyttäjälle install.php-tiedoston.

Select an installation profile



☒ **Standard**  
Install with commonly used features pre-configured.

☐ **Minimal**  
Start with only a few modules enabled.

► **Choose profile**

Choose language

Verify requirements

Set up database

Install profile

Configure site


Finished

Save and continue

## KUVA 6. Asennussivu

Aivan aluksi valitaan, halutaanko standardi- vai minimaalinen asennus. Tässä työssä valitaan standardi-asennus. Seuraavaksi valitaan kieli, jota drupal käyttää admin-näkymässä. Oletuksena kieleksi tarjotaan englantia, jota käytetään tässäkin työssä, tai ohjeet miten muita kieliä saadaan käyttöön. Tämän jälkeen Drupal tarkistaa, onko palvelin järjestelmävaatimusten mukainen. Ellei palvelin täytä vaatimuksia, Drupal palauttaa sivun, jossa korjattavan asian näkee suoraan.

Requirements problem



✓ Choose profile

✓ Choose language

► **Verify requirements**

Set up database

Install profile

Configure site

Finished

Web server	Apache
PHP	5.2.6
PHP register globals	Disabled
PHP extensions	Enabled
Database support	Enabled
PHP memory limit	64M
<div style="background-color: #f0f0f0; padding: 5px;"> <p>✗ <b>File system</b></p> <p>The directory <code>sites/default/files</code> does not exist. An automated attempt to create this directory failed, possibly due to a permissions problem. To proceed with the installation, either create the directory and modify its permissions manually or ensure that the installer has the permissions to create it automatically. For more information, see <a href="#">INSTALL.txt</a> or the <a href="#">online handbook</a>.</p> </div>	
Unicode library	PHP Mbstring Extension
<div style="background-color: #f0f0f0; padding: 5px;"> <p>✗ <b>Settings file</b></p> <p>The settings file does not exist.</p> <p>The Drupal installer requires that you create a settings file as part of the installation process. Copy the <code>./sites/default/default.settings.php</code> file to <code>./sites/default/settings.php</code>. More details about installing Drupal are available in <a href="#">INSTALL.txt</a>.</p> </div>	

Check the error messages and [proceed with the installation](#).

## KUVA 7. Ongelma järjestelmävaatimuksissa

Järjestelmävaatimuksien täyttyessä Drupal tarvitsee tietokantatiedot, kuten: Tietokantatyypin, tunnukset sekä kannan nimen. Tämän jälkeen Drupal luo tietokantaan tarvittavat taulut perusasetuksineen.

Configure site

☒ Choose profile  
☒ Choose language  
☒ Verify requirements  
☒ Set up database  
☒ Install profile  
☒ **Configure site**  
 Finished

**SITE INFORMATION**

Site name \*

Site e-mail address \*

Automated e-mails, such as registration information, will be sent from this address. Use an address ending in your site's domain to help prevent these e-mails from being flagged as spam.

**SITE MAINTENANCE ACCOUNT**

Username \*

Spaces are allowed; punctuation is not allowed except for periods, hyphens, and underscores.

E-mail address \*

Password \* Password strength:

Confirm password \*

**SERVER SETTINGS**

Default country

Select the default country for the site.

Default time zone

By default, dates in this site will be displayed in the chosen time zone.

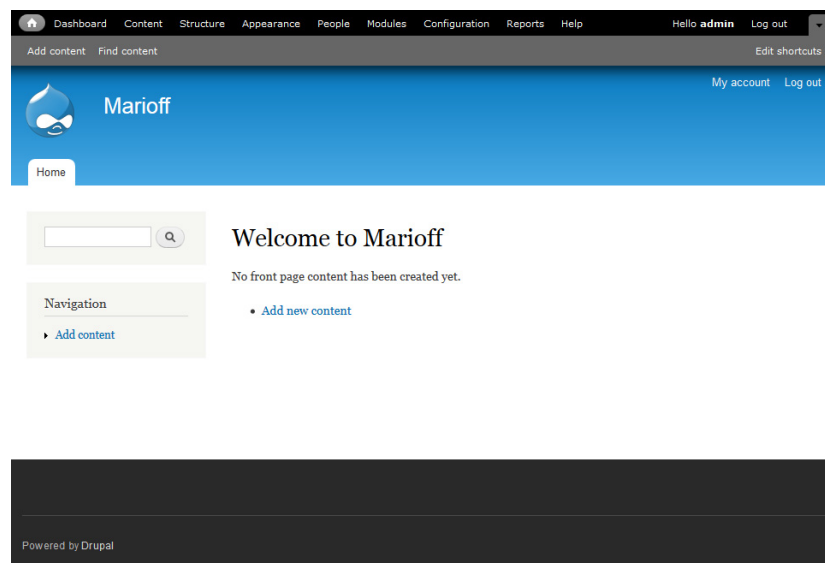
**UPDATE NOTIFICATIONS**

☒ Check for updates automatically  
☐ Receive e-mail notifications

The system will notify you when updates and important security releases are available for installed components. Anonymous information about your site is sent to [Drupal.org](http://Drupal.org).

## KUVA 8. Sivuston konfigurointi

Tämän jälkeen annetaan sivun nimi, pääkäyttäjän tunnukset, aika-asetukset sekä valittavana on, halutaanko päivityksiä tarkistaa automaattisesti ja lähetetäänkö pääkäyttäjälle sivustolta sähköposteja. Asennuksen jälkeen ohjataan pääsivulle, joka näyttää tältä:



KUVA 9. Asennus valmis

## 4.2 Moduulien asennus

Koska Drupaliin on valmiiksi koodattu todella paljon lisämoduuleita, kannattaa niitä käyttää. Näin säästyy aikaa, koska kaikkea ei tarvitse itse tehdä alusta asti. Tässä muutama moduuli, joita projektissa on käytössä:

**Devel:** Kehitysmoduuli. Mahdollistaa paremman debuggauksen. Esimerkiksi `dpm()`-funktioilla saadaan tulostettua sivulle muuttujat selattavassa taulukossa.

**Media:** Tarjoaa laajennetut puitteet mediatiedostojen käsittelyyn.

**Menu block:** Luo konfiguroitavia blokkeja valikoista, millä voi vaikuttaa esimerkiksi siihen, kuinka monta alatasoa näytetään ja miltä tasolta aloitetaan.

**Node weight:** Mahdollistaa nodejen järjestelyn painoarvon mukaan.

**Ckeditor:** Mahdollistaa CKEditorin ( <http://ckeditor.com/> ) käyttämisen drupalin tekstikentissä.

**Token:** Token moduuli tarjoaa keskitetyn API:n käyttäen placeholdereita, joita korvataan tekstillä. Esimerkiksi [user] token saisi arvoksi käyttäjänimen, kuka sivua silloin käyttää.

**Pathauto:** Moduuli parempien url-osoitteiden rakentamiseen. Oletuksena Drupalissa sisältöön pääsee polulla node/nid, missä nid on noden id-arvo. Pathauton avulla polun voi rakentaa tokeneiden avulla esimerkiksi sisältötyyppi/node-title muotoisen aliaksen, jolloin osoitteet selkiytyvät huomattavasti. Moduuli vaatii token moduulin toimiakseen.

**Views:** Sisällön listaamiseen tarkoitettu moduuli, joka ei vaadi koodausta vaan asetukset ja parametrit voi säätää suoraan käyttöliittymästä. Voi luoda listaussivuja tai listausblokkeja.

Moduulit voi asentaa suoraan käyttöliittymästä klikkaamalla moduulit-sivulta löytyvää asenna moduuli -linkkiä tai siirtämällä moduulikansio suoraan palvelimelle sites/default/modules-kansioon.

### 4.3 Sisältötyyppien luonti

Referenssit tarvitsevat oman sisältötyypin sisällön luontia varten. Referenssit vaativat pikkukuvan, joka näytetään listauksessa, sekä taustakuvan, joka näkyy referenssiä katsottaessa. Myös tekstikenttä vaaditaan kuvauksen kirjoittamiseen, mihin oletuksena sisältötyyppeihin tuleva body-kenttä käy hyvin.

Sisältötyyppien luonti tapahtuu menemällä admin-valikosta *Structure* ja *Content types*. Hallintanäkymä on samanlainen kuin esimerkiksi blokeilla. Vasemmalla on sisältötyypin nimi, machine name eli koneluettava nimi sekä mahdollinen kuvaus ja oikealla on toimenpiteet: muokkaa, hallinnoi kenttiä, ylläpidä näkymää sekä poista. Uuden sisältötyypin luonti tapahtuu taulukon päällä vasemmalla olevasta create new content type -linkistä.

Sisältötyypin luontisivulla pakollisia kenttiä ovat sisältötyypin nimi sekä otsikkokentän nimi, jotka ovat tässä työssä reference sisältötyypin nimeksi ja otsikoksi title. Nimeä kirjoittaessa Drupal tekee sisällölle samaan aikaan myös machine

namen eli koneluettavan nimen, joka tallentuu noden *type* propertyksi, jota voidaan käyttää koodissa esimerkiksi rajattaessa nodeja sisältötyypin perusteella.

Kuville kentät luodaan käyttäen image-kenttätyyppiä ja media file selector widgettiä.

KUVA 10. Media file selector widget

#### 4.4 Taksonomiat

Marioff-sivuston referenssit on jaettu kolmeen kategoriaan: buildings, industry & energy ja marine & offshore. Jotta referenssit voidaan liittää johonkin kategoriaan, tarvitsee näille luoda taksonomia. Taksonomia hallinta löytyy valikosta *structure/taxonomy*, ja tänne luodaan sanasto nimeltään reference category. Tämän alle luodaan kategorian vaatimat termit, jotka voidaan valita referenssinodea luotaessa.

Taksonomian luonnin jälkeen se täytyy liittää referenssi nodeen. Referenssinodeen lisätään uusi kenttä *field\_category*, jonka tyyppiä annetaan term reference sekä widgetiksi select list. Termin referenssiksi annetaan luotu reference category -sanasto. Tämä luo referenssi sisältötyyppiin select-tyyppisen html-elementin reference category -sanaston termeistä.

LABEL	MACHINE NAME	FIELD TYPE	WIDGET	OPERATIONS
+ Language	language	Language selection		
+ Title	title	Node module element		
+ Image	field_image	Image	Media file selector	edit delete
+ Small image	field_small_image	Image	Media file selector	edit delete
+ Category	field_category	Term reference	Select list	edit delete
+ Body	body	Long text and summary	Text area with a summary	edit delete

## KUVA 11. Katgoria lisättynä

### 4.5 Moduulin luonti

Tässä työssä luodaan marioff reference -moduuli, joka hoitaa kaikki referenssi-gallerian tarvitseman logiikan, kuten tietokantahaut ja blokit.

Moduulit luodaan sites/default/modules-hakemistoon ja jokainen moduuli luodaan omaan hakemistoonsa. Moduulit tarvitsevat .info- ja .module-tiedostot toimiakseen ja näkyäkseen Drupalille. Moduulin tiedot ja metadata menevät .info-tiedostoon ja ohjelmakoodit .module-tiedostoon. Tässä tapauksessa marioff\_references-hakemistoon luotiin marioff\_references.info- sekä marioff\_references.module-tiedostot.

Marioff\_references.info-tiedosto:

```
name = Marioff references
description = Marioff references module
core = 7.x
package = VJ
```

Tämän jälkeen .module-tiedostoon koodataan tarvittava logiikka. Ensimmäisenä hook\_menu, jolla päästään käsiksi dataan, kun käyttäjä menee tiettyyn osoitteeseen, joka tässä tapauksessa on references.

```
/*
 * implements hook_menu()
 */

function marioff_references_menu() {

    $items['references'] = array(
```

```

        'file' => 'marioff_references.pages.inc',
        'page callback' => 'marioff_references_references_page',
        'page arguments' => array(1),
        'access arguments' => array('access content'),
        'type' => MENU_CALLBACK,

    );

    return $items;
}

```

Koska moduulin nimi on marioff\_references ja käytettävä hookki on hook\_menu, saadaan funktioksi marioff\_references\_menu(). Funktio palauttaa \$items-taulukon, johon on määritelty:

- File, tiedosto, mistä funktio löytyy
- Page callback, funktio, jota kutsutaan, kun käyttäjä menee referensseihin
- Page arguments, argumentit, jotka lähetetään funktiolle, tässä tapauksessa references/ jälkeen tuleva url
- Access arguments, käyttöäoikeudet
- Type, valikkokohdan tyyppi, tässä tapauksessa MENU\_CALLBACK, joka rekisteröi url polun.

Käyttäjän mennessä references-osoitteeseen kutsutaan marioff\_references\_references\_page() funktiota, joka sijaitsee marioff\_references.pages.inc-tiedostossa. Tämä funktio ottaa vastaan parametrina mahdollisen valitun kategorian. Jos kategoriaa ei ole valittuna, ohjataan käyttäjä ensimmäisen kategorian ensimmäiseen referenssiin. Jos kategoria on valittuna, ohjataan käyttäjä kyseisen kategorian ensimmäiseen referenssiin.

Määritellään funktio, joka ottaa kategorian vastaan parametrina, joka on oletuksena false, jos tullaan suoraan references osoitteeseen.



```

function marioff_references_references_page($cat =
false) {

    if(!$cat){

        $tid =
        _marioff_references_get_first_tid('reference_ca
tegrory');

    } else {

        //get tid from url
        $url_suggestion = 'reference-page/' . $cat; //
taxonomy pattern
        $orig = dru-
pal_get_normal_path($url_suggestion);
        $path_parts = explode('/', $orig);
        $tid = end($path_parts);

    }

    $first_ref =
    _marioff_references_get_first_reference($tid);
    $node = node_load($first_ref->nid);

    // get node path
    $path = dru-
pal_lookup_path('alias',"node/".$node->nid);

    //redirect user
    drupal_goto($path);

}

```

Kun käyttäjä saapuu suoraan referenssisivulle, haetaan ensimmäinen termi referenssikategoria-sanastosta kustomifunktiolla `_marioff_references_get_first_tid()`, joka palauttaa annetun sanaston ensimmäisen termin id:n.

```
function _marioff_references_get_first_tid($vocabulary)
{
    $voc = taxonomo-
my_vocabulary_machine_name_load($vocabulary);
    $tree = taxonomy_get_tree($voc->vid);
    $tid = $tree[0]->tid;
    return $tid;
}
```

Funktio ottaa vastaan parametrina sanaston koneluettavan nimen, jonka avulla ladataan sanasto. Sanaston termit ladataan \$tree-muuttujaan, josta otetaan taulukon ensimmäinen alkio, jonka tid property palautetaan.

Mikäli käyttäjä on valinnut kategorian haetaan termin id osoitteen perusteella.

```
$url_suggestion = 'reference-page/' . $cat;
```

Koska taksonomioille on määritelty url-aliakset muotoon reference-page/termin nimi, saadaan tama tallennettua \$url\_suggestion-muuttujaan.



▼ **URL PATH SETTINGS** (AUTOMATIC ALIAS)

☒ Generate automatic URL alias  
 Uncheck this to create a custom alias below. [Configure URL alias patterns.](#)

**URL alias**

Optionally specify an alternative URL by which this term can be accessed. Use a relative path and don't add a trailing slash or the URL alias won't work.

KUVA 12. URL-alias määritettynä termille

```
$orig = drupal_get_normal_path($url_suggestion);
```

Tämän jälkeen voidaan hakea taksonomian normaali url funktiolla `drupal_get_normal_path()` ja tallentaa se muuttujaan. Funktio palauttaa stringin muodossa `taxonomy/term/3`, missä viimeinen numero on termin id-numero.

Kun kategorian termin id on tiedossa, voidaan sen avulla hakea tietokannasta ensimmäinen referenssi kyseisestä kategoriasta. Tätä varten tehtiin kustomifunktio `_marioff_references_get_first_reference($tid)`, joka ottaa vastaan termin id:n.

```
function _marioff_references_get_first_reference($tid =
false) {

    if($tid){

        $query = new EntityFieldQuery();

        $query
        ->entityCondition('entity_type', 'node')
        ->entityCondition('bundle', 'reference')
        ->fieldCondition('field_category', 'tid', $tid,
        '=')
        ->fieldOrderBy('field_node_weight', 'value',
        'ASC')
        ->range(0,1);

        $result = $query->execute();
        $reference = reset($result['node']);

        return $reference;

    } else {

        return false;

    }
}
```

Funktiossa on käytetty Drupalin omaa entity field query -luokkaa, jolla tietokannasta voi hakea dataa erilaisten ehtojen avulla. Tässä tapauksessa haetaan etiteettiä

tyypillä node, joka kuuluu tyyppiin reference. Kentistä tarkastatetaan, että field\_category vastaa annettua termi id:tä. Järjestys perustuu ennaltamääritettyyn painoarvoon, joka on määritelty erillisellä node weight -moduulilla, joka luo tarville nodeille painoarvokentän, jota voi säätää. Tietokannasta otetaan vain ensimmäinen tulos ja tämä kysely ajetaan sekä saatu taulukko tallennetaan \$result-muuttujaan. Vielä varmuuden vuoksi tallennetaan \$reference muuttujaan \$result-tilukon node-avaimen ensimmäinen alkio, missä on node id, ja tämä palautetaan.

Kun kategorian ensimmäinen node id on tiedossa, voidaan node ladata sekä hakea osoitealias kyseiselle nodelle ja ohjata käyttäjä sinne.

```
// get node path
$path = drupal_lookup_path('alias',"node/".$node-
>nid);

//redirect user
drupal_goto($path);
```

## 4.6 Blokit

Kirjautumalla admin-tunnuksilla sisään pääsee blokkeihin käsiksi menemällä ensin admin-valikosta *structures* ja avautuvalta sivulta valitaan *blocks*. Kyseinen sivu on blokkien hallintasivu ja ylhäällä oikealla näkyvät tällä hetkellä aktivoidut teemat, joista tällä hetkellä aktiivinen teema on valmiiksi valittuna. Tämä tarkoittaa sitä, että blokkien asetukset vaikuttavat vain valitussa teemassa, eli eri asetuksia pystyy säätämään teemojen välillä.

Hallintasivun taulukossa pystyy säätämään, millä alueella blokki halutaan näkyvän ja missä järjestyksessä blokit halutaan tulostaa. Blokkeja voi raahata paikoilleen drag & drop -menetelmällä tai *region*-sarakkeen alavetovalikosta voi valita suoraan oikean alueen. Operations-sarakkeesta sitten pääsee kyseisen blokin asetuksiin käsiksi tai sieltä voi myös poistaa omia blokkejaan.

Blokkien tarkempi konfigurointi tapahtuu blokin perässä olevasta configure-linkistä, josta pääsee rajaamaan tarkemmin blokin näkyvyyttä. Ensiksi tällä sivulla pääsee antamaan blokkille otsikon tai sitten syöttämällä kenttään <none> voi otsikon piilottaa. Seuraavaksi voi valita teemoittain blokkille alueen, johon se halutaan sijoittaa. Viimeisenä on visibility settings, josta pääsee rajaamaan näkyvyyttä sivujen-, käyttäjätyyppien- tai käyttäjäroolien mukaan sekä voi asettaa, onko blokki käyttäjien kustomoitavissa vai ei.

Referenssigalleriaan tarvitaan kaksi blokkia:

- Kategorialistaus, jonka avulla käyttäjä voi valita selattavan kategorian
- Referenssilistaus, missä näytetään kyseisen kategorian referenssit.

Kategorialistaus tulee taksonomiatermistä, ja referenssilistauksen tekemiseen voi käyttää views-moduulia.

Blokkien tekemiseen tarvitaan kahta hookkia: Hook\_block\_info ja hook\_block\_view. Hook\_block\_info kertoo drupalille, mitä blokkeja moduuli tarjoaa ja mitä konfigurointeja sillä on. Hook\_block\_view vastaa sisällön generoimisesta.

```
function marioff_references_block_info() {

  // category submenu block
  $blocks['cat_list'] = array(

    'info' => t('References category submenu'),
    'cache' => DRUPAL_CACHE_GLOBAL

  );

  // reference listing block

  $blocks['ref_list'] = array(
```

```

        'info' => t('References listing block'),
        'cache' => DRUPAL_CACHE_GLOBAL

    );

    return $blocks;

}

```

Blokkien koneluettavat nimet määritellään \$blocks-taulukon avaimiksi ja käyttäjille näkyvä nimi määritellään info-kohtaan. Cachetusvaihtoehtoja on useampia, mutta nyt käytössä on DRUPAL\_CACHE\_GLOBAL, mikä tarkoittaa sitä, että blokki on kaikille sama, missä se on näkyvissä.

Hook\_block\_viewissä funktion argumentiksi tulee hook\_block\_infon määrittelemä blokin koneluettava nimi. Funktion rakenne on yleensä switch case-ehtolauseen muodossa, missä yksittäinen case vastaa yhtä blokkia.

```

function marioff_references_block_view($delta = '') {

    $block = array();
    $cn = menu_get_object();
    $cat = field_get_items('node', $cn,
        'field_category');

    switch ($delta) {

        case 'cat_list':

            break;

        case 'ref_list':

            break;

    }

    return $block;
}

```

```
}
```

Tässä on perus hook\_block\_view rakenne, aluksi haetaan käyttäjän nykyinen node muuttujaan \$cn sekä kyseisestä nodesta kategoriatieto field\_category-kentästä ja tallennetaan se \$cat-muuttujaan.

### **Kategorialistausblokki:**

```
case 'cat_list':

if(!empty($cat[0]['tid'])) {

    // category term listing

    $voc = taxono-
my_vocabulary_machine_name_load('reference_category
');
    $tree = taxonomy_get_tree($voc->vid);

    foreach ($tree as $term) {

        $term = taxonomy_term_load($term->tid);
        $url = dru-
pal_get_path_alias('taxonomy/term/'.$term->tid);
        $url = str_replace('reference-page/', 'refer-
ences/', $url);

        //css attributes
        $attributes = array('title' => $term->name);
        if($term->tid == $cat[0]['tid']) {

            $attributes['class'] = array('active');

        }
    }
}
```

```

        // items array() link tekst, link, css
        $items[] = l($term->name, $url, array('attributes' => $attributes));
    }

    // ul list render array

    $block['content'] = array(

        '#theme' => 'item_list',
        '#items' => $items,
        '#attributes' => array('id' => 'category-list')

    );

    $block['subject'] = '';

}

break;

```

Aluksi haetaan kaikki reference category-sanaston termit ja käydään ne läpi foreach-loopilla, missä jokaisesta termistä muodostetaan linkki Drupalin l funktion avulla. Näistä linkit lisätään numeroimattomaan luetteloon, jonka id:ksi annetaan category-list.

### **Referenssilistausblokki:**

```

case 'ref_list':

    if(!empty($cat[0]['tid'])) {

        $reference_list =
        views_embed_view('reference_listing',
        'ref_list', $cat[0]['tid']);

        $block['content'] = array(

```



```

        '#markup' => $reference_list

    );

    $block['subject'] = '';

}

break;

```

Koska referenssilistausblokki on tehty views-moduulin avulla, täytyy sen sisältö hakea käyttäen apuna `views_embed_view`-funktia, jonka argumentiksi annetaan viewin nimi, näkymän nimi sekä kategorian termi-id.

The image shows the Drupal Views configuration interface for a view named 'reference listing'. The interface is split into two panes.

**Left Pane (Main Configuration):**

- TITLE:** Title: `reference listing`
- FORMAT:** Format: `Grid` | `Settings`; Show: `Fields` | `Settings`
- FIELDS:** (Add button)
  - Field: `Image`
  - Content: `Small image`
  - Content: `Title`
- FILTER CRITERIA:** (Add button)
  - Content: `Published (Yes)`
  - Content: `Type (= Reference)`
  - Content: `Language (= Current user's language)`
- SORT CRITERIA:** (Add button)
  - Content: `Node weight (asc)`

**Right Pane (Advanced Settings):**

- CONTEXTUAL FILTERS:** (Add button)
  - Content: `Category`
- RELATIONSHIPS:** (Add button)
- NO RESULTS BEHAVIOR:** (Add button)

KUVA 13. Reference listing viewsin asetukset

Kun lohkot on luotu, täytyy ne vielä käydä sijoittamassa oikeille alueille sekä konfigurointia täytyy muuttaa niin, että blokit näkyvät vain referenssinodeissa. Katteoria- ja listausblokit sijoitetaan content-alueelle ja blokin näkyvyysasetuksista rajoitetaan näkyvyys sisältötyypin perusteella näkymään vain referenssinodeissa.

## 4.7 Teeman luonti

Luodaan kansio *marioff* kansioon *sites/default/themes*. Tähän kansioon tulevat kaikki marioff-teemaan liittyvät tiedostot, kuten .info-tiedosto, templatet ja css. Ainoa vaadittu tiedosto on .info-tiedosto, jossa määritellään ja konfiguroidaan teema.

### 4.7.1 .info-tiedosto

Drupal käyttää .info-tiedostoja metadatan säilyttämiseen moduuleilla ja teemoilla. Tiedostoon pystytään määrittelemään esimerkiksi teeman nimi, kuvaus, versio, lohkojen alueet, css-tiedostot, javascript-tiedostot sekä muuta dataa. Lisätietoa löytyy Drupalin omasta documentaatiosta <https://drupal.org/node/171205>.

Teeman koneluettava nimi määräytyy myös .info-tiedoston nimen mukaan, joten tässä tapauksessa teema-kansioon tulee marioff.info ja näin ollen teemaan voi viitata PHP:ssä nimellä *marioff*. (Drupal 2014c).

Tiedosto käyttää formaattia, joka on `key = value` eli esimerkiksi:

```
name = Marioff
```

Pakollisia parametreja .info-tiedostossa ovat `name` ja `core`:

**name:** nimi, joka näkyy Drupalin teemanvalintasivulla.

```
name = Marioff
```

**description:** moduulin lyhyt kuvaus, joka kertoo adminille, mitä moduuli tekee. Maksimi merkkimäärä on 255 merkkiä.

```
description = Marioff theme
```

**core:** Drupalin versio, johon moduuli on ohjelmoitu. Drupalin versioon 7 käytetään muotoa `7.x` ja versioon 6 käytetään `6.x`.

```
core = 7.x
```

**stylesheets:** teeman tarvitsemat css-tiedostot voidaan määritellä tähän sekä media-tyyppi, esimerkiksi:

```
stylesheet[all][] = style.css
```

PHP:n vuoksi käytetään hakasulkeita, jotta saadaan data taulukoksi ja tyhjät hakasulkeet loppuun, jotta data menee taulukossa seuraavaan tyhjään paikkaan.

Tässä tapauksessa lisätään reset.css tyylien resetoimiseen, layout.css yleisille tyy-  
leille sekä typo.css typograafisille tyy-  
leille.

```
stylesheets[all][] = css/reset.css  
stylesheets[all][] = css/layout.css  
stylesheets[all][] = css/typo.css
```

**scripts:** Tässä määritellään teeman tarvitsevat script tageissa tulevat tiedostot.

Tässä tapauksessa marioff.js yleisille javascripteille sekä superfish.js kirjasto navigaatiolle.

```
scripts[] = js/marioff.js  
scripts[] = js/superfish.js
```

**Regions:** Teeman regionien eli alueiden määrittelemiseen. Laitetaan muodossa  
region[regionin\_koneluettava\_nimi] = Regionin nimi.

Tässä tapauksessa regionit näyttävät tältä:

```
regions[header] = Header  
regions[content] = Content  
regions[sidebar_first] = Sidebar first  
regions[bottom] = Page bottom  
regions[footer] = Footer
```

Lopputulokseksi saatiin .info-tiedosto näillä tiedoilla:

```
name = Marioff
```

```

description = Marioff theme
core = 7.x

stylesheets[all][] = css/reset.css
stylesheets[all][] = css/layout.css
stylesheets[all][] = css/typo.css
stylesheets[all][] = css/superfish.css

scripts[] = js/marioff.js
scripts[] = js/superfish.js

regions[header] = Header
regions[content] = Content
regions[sidebar_first] = Sidebar first
regions[bottom] = Page bottom
regions[footer] = Footer

```

#### 4.7.2 Template.php

Tempalte.php on tiedosto joka lähes aina löytyy teeman juuresta ja joka on tarkoitettu teemafunktioiden ylikirjoittamiseen (Hodhdon 2013, 12).

Teemasta halutaan piilottaa muropolku, kun ollaan referenssigalleriassa, ja jos nodessa on asetettu taustakuva, täytyy se laittaa paikalleen. Tähän toimii funktio `theme_preprocess_page`, joka käsittelee sivun muuttujia. Koska teeman nimi on `marioff`, muodostuu funktion nimeksi `function marioff_preprocess_page()`.

```

function marioff_preprocess_page(&$vars) {

    $node = isset($vars['node']) ? $vars['node'] :
    false;

    if($node->type == 'reference') {

        $vars['breadcrumb'] = FALSE;
    }
}

```

```

$image = field_get_items('node', $node,
'field_image');

if(isset($image[0]['uri'])) {

    $vars['main_bg'] =
        file_create_url($image[0]['uri']);

}

}

```

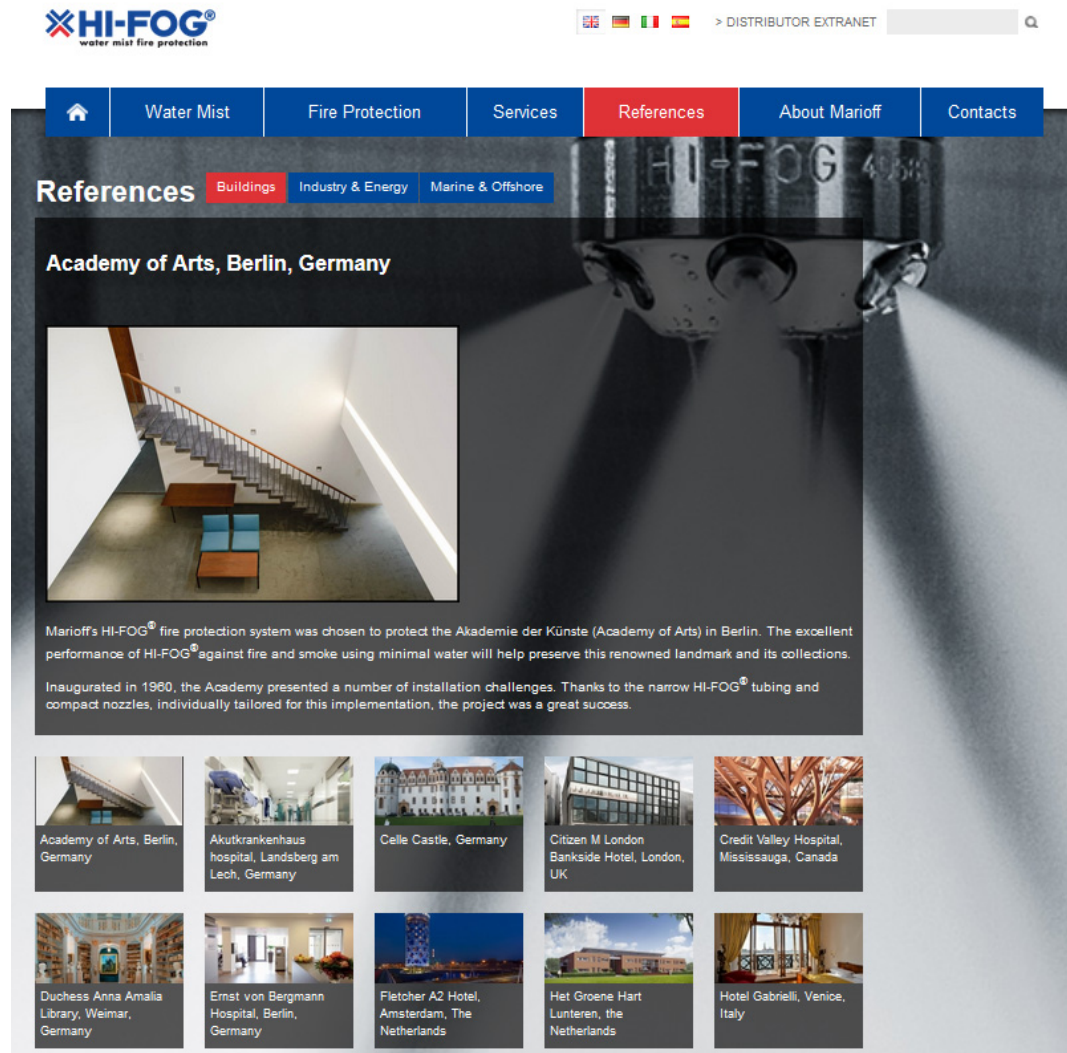
Mikäli ollaan selaamassa nodea löytyy se \$vars taulukon avaimesta ['node'], tämä tallennetaan \$node-muuttujaan. Jos node on tyyppiä referenssi, asetetaan breadcrumb falseksi, jolloin se ei enää tulostu. Kenttien hakemiseen node-objektista löytyy funktio field\_get\_items, joka palauttaa kentän tiedot taulukkomuodossa. Kuvan paikkatieto löytyy kohdasta ['uri'], joka on muotoa public://tiedoston polku, missä public:// on Drupalin asetuksissa määritelty julkinen tiedostokansio. Jotta tästä saataisiin muodostettua polku, jota voidaan käyttää CSS:ssä, tarvitsee se käyttää file\_create\_url-funktion läpi, joka palauttaa kuvan oikean polun. Polku tallennetaan \$vars-tilin main\_bg-avaimen, joka on suoraan käytettävissä muuttujana \$main\_bg page.tpl.php-tiedostossa.

```

<div id="main" class="clearfix" <?php isset($main_bg) ?
print 'style="background:url('.$main_bg.');"' : ''; ?>
>

```

CSS-tyylien lisäyksen jälkeen referenssigalleria on valmiina käyttöön ja opinnäytetyötä tehdessä näyttää tältä:



KUVA 14. Valmis referenssigalleria

## 5 YHTEENVETO

Opinnäytetyössä käsiteltiin yleisesti sisällönhallintajärjestelmiä ja vielä tarkemmin [www-sisällönhallintajärjestelmiä](#). Drupal otettiin tarkempaan tutkiskeluun ja siitä käytiin läpi yleistä toimintaa sekä perusteita ja käsitteitä.

Case-osuudessa toteutettiin referenssigalleria Marioffin sivulle koodaamalla sitä varten oma moduuli. Moduulin rakenne käytiin läpi tarkoituksena ottaa teoriaosuuden asiat käyttöön koodiesimerkkien avulla.

Referenssigalleria onnistui hyvin ilman suurempia ongelmia. Muitakin erilaisia toteutustapoja gallerialle olisi ollut ja asioita olisi voinut tehdä erilailla, mikä on Drupalin vahvuuksia, ettei kaikkia asioita tarvitse tehdä samaan tyyliin. Hyvin tärkeää on myös osata käyttää Drupalin api-sivua hyväkseen. Hookeja ja muita funktioita on niin paljon, että niitä on vaikea yrittää muistaa. Kannattaa katsoa Drupalin sivuilta tarvittavat tiedot, missä ne on kattavasti dokumentoituna. Myös valmiita moduuleja löytyy valtavasti, ettei kaikkea tarvitse itse alusta asti ruveta koodaamaan.

## LÄHTEET

### PAINETUT LÄHTEET

Hodhdon, J. 2013. Programmer's Guide to Drupal. California: O' Reilly Media, Inc.

Melançon, B., Luisi, J., Négyesi, K., Anderson, G., Somers, B., Corlosquet, S., Freudenberg, S., Lauer, M., Carlevale, E., Lorétan, F., Nordin, D., Szrama, R., Stewart, S., Strawn, J., Travis, B., Hakimzadeh, D., Scavarda, A., Albala, A., Micka, A., Douglass, R., Monks, R., Scholten, R., Wolanin, P., VanValkenburgh, K., Stout, G., Qynn Dolin, K., Gifford, M., Sarahe, C., Boyer, S. & Mars, F. 2011. The Definitive Guide to Drupal 7. New York: Apress Media LLC.

Shreves, R. & Dunwoodie, B. 2011. Drupal 7 Bible. Indiana: Wiley Publishing, Inc.

### SÄHKÖISET LÄHTEET

Drupal, 2014a. Drupal [viitattu 08.03.2014]. Saatavissa:  
<https://drupal.org/>

Drupal, 2014b. Fields [viitattu 26.03.2014]. Saatavissa:  
<https://drupal.org/node/1577064>

Drupal, 2014c. Overview of theme files [viitattu 03.03.2014]. Saatavissa:  
<https://drupal.org/node/171194>

Drupal, 2014d. System requirements [viitattu 18.02.2014]. Saatavissa:  
<https://drupal.org/requirements>

Drupal, 2014e. The Drupal overview [viitattu 18.02.2014]. Saatavissa:  
<https://drupal.org/getting-started/before/overview>

Drupal, 2014f. Working with blocks [viitattu 18.02.2014]. Saatavissa:  
<https://drupal.org/documentation/modules/block>



Drupal, 2014g. Writing theme .info files [viitattu 03.03.2014]. Saatavissa:

<https://drupal.org/node/171205>

Wikipedia, 2014a. Content management system [viitattu 08.03.2014]. Saatavissa:

[http://en.wikipedia.org/wiki/Content\\_management\\_system](http://en.wikipedia.org/wiki/Content_management_system)

Wikipedia, 2014b. GNU General Public License [viitattu 08.03.2014]. Saatavissa:

[http://en.wikipedia.org/wiki/GNU\\_GPL](http://en.wikipedia.org/wiki/GNU_GPL)

Wikipedia. 2014c. Sisällönhallintajärjestelmä [viitattu 08.03.2014]. Saatavissa:

<http://fi.wikipedia.org/wiki/Sis%C3%A4ll%C3%B6nhallintaj%C3%A4rjestelm%C3%A4>

W3techs. 2014. Usage of content management systems for websites [viitattu 19.03.2014]. Saatavissa:

[http://w3techs.com/technologies/overview/content\\_management/all](http://w3techs.com/technologies/overview/content_management/all)

Websitesetup. 2014 CMS COMPARISON - WordPress, Drupal & Joomla [viitattu 22.03.2014]. Saatavissa:

<http://websitesetup.org/cms-comparison-wordpress-vs-joomla-drupal/>

Verkkojulkaisut. 2014. Yritys [viitattu 22.3.2014]. Saatavissa:

<http://www.verkkojulkaisut.fi/olemmeyksi-suomen-kokeneimmista-digitoimistoista>

## KUVALÄHTEET

Drupal, 2014e. The Drupal overview [viitattu 18.02.2014]. Saatavissa:

<https://drupal.org/getting-started/before/overview>

Hodhdon J. 2013. Programmer's Guide to Drupal. California: O'Reilly Media, Inc