



Building a webshop using Drupal Commerce

with Commerce Kickstart as a Template

Axel Wiklund

Degree Thesis
Information Technology
2015

DEGREE THESIS	
Arcada	
Degree Programme:	Information Technology
Identification number:	4804
Author:	Axel Wiklund
Title:	Building a webshop using Drupal Commerce with Commerce Kickstart as a Template
Supervisor (Arcada):	Jonny Karlsson
Commissioned by:	Genero Oy Ab
<p>Abstract:</p> <p>This thesis deals with the implementation of a Drupal Commerce based web shop in Drupal. The objective is to describe how to implement the Drupal Commerce framework using Commerce Kickstart as a model based on practical experience gained from building a web site for a fitness company, called Fitspire. Drupal Commerce is an e-commerce framework for Drupal. Commerce Kickstart is an installation profile for Drupal Commerce containing the modules and configurations needed to set up a basic Drupal Commerce shop.</p> <p>The theoretical part of the thesis will compare the most known CMSs and explain the reasoning behind choosing Drupal as a platform for a web shop. It will then compare a few available e-commerce solutions out there and motivate why Drupal Commerce is an optimal choice for building advanced web shops.</p> <p>The practical part of the thesis deals with the different steps involved in implementing the Drupal Commerce framework and setting up a web shop. The Fitspire project is used as a source of examples for the different steps.</p>	
Keywords:	Drupal, commerce, commerce kickstart, web shop
Number of pages:	
Language:	English
Date of acceptance:	

EXAMENSARBETE	
Arcada	
Utbildningsprogram:	Informationsteknologi
Identifikationsnummer:	4804
Författare:	Axel Wiklund
Arbetets namn:	Building a webshop using Drupal Commerce with Commerce Kickstart as a Template
Handledare (Arcada):	Jonny Karlsson
Uppdragsgivare:	Genero Oy Ab
<p>Sammandrag:</p> <p>Detta examensarbete handlar om implementering av en webbshop med Drupal Commerce i Drupal. Syftet med arbetet är att beskriva hur man implementerar Drupal Commerce genom att använda Commerce Kickstart som modell. Beskrivningen är baserad på praktiskt erfarenhet från utveckling av en webbsida för ett träningsföretag, kallat Fit-spire. Drupal Commerce är ett e-handelsramverk för Drupal. Commerce Kickstart är en installationsprofil för Drupal Commerce, som innehåller de moduler och inställningar som behövs för att skapa en simpel Drupal Commerce webbshop.</p> <p>Den teoretiska delen av detta arbete jämför de mest kända innehållshanteringssystemen och motiverar valet av Drupal som plattform för en webbshop. Det jämför sedan några olika e-handelslösningar och motiverar varför just Drupal Commerce är ett optimalt val för att bygga en avancerad webbshop.</p> <p>Den praktiska delen av examensarbetet behandlar de olika stegen involverade i implementering av Drupal Commerce ramverket och uppbyggandet av en webbshop. Fit-spire-projektet används som en källa för exempel för de olika stegen.</p>	
Nyckelord:	Drupal, commerce, commerce kickstart, webbshop
Sidantal:	
Språk:	Engelska
Datum för godkännande:	

CONTENTS

Introduction.....	8
1.1 Background and Scope	8
1.2 Objective and Methods.....	8
1.3 Thesis Structure	9
2 Content Management Systems.....	10
2.1 WordPress.....	10
2.2 Joomla	11
2.3 Drupal	12
2.4 Summary	13
3 E-Commerce Solutions	15
3.1 Magento.....	15
3.2 Ubercart.....	15
3.3 Drupal Commerce	16
3.3.1 <i>Commerce Kickstart</i>	16
3.4 Summary	17
4 Building a Commerce Site.....	19
4.1 Installing Drupal Commerce	19
4.1.1 <i>Drupal Core Installation</i>	19
4.1.2 <i>Drush</i>	20
4.1.3 <i>Setting up Drupal Commerce</i>	20
4.2 Creating the Products.....	22
4.2.1 <i>Nodes</i>	22
4.2.2 <i>Product Variations</i>	23
4.3 Displaying the Products.....	24
4.3.1 <i>Display Suite</i>	24
4.3.2 <i>Views</i>	25
4.3.3 <i>Facets</i>	27
4.4 The Checkout Process	29
4.4.1 <i>Shipping Costs</i>	30
4.4.2 <i>Payment Systems</i>	31
4.5 Order Notifications.....	34
4.5.1 <i>Rules</i>	34
4.5.2 <i>Notifications</i>	35

4.6	Product Recommendations	36
4.7	Store Management	37
4.7.1	<i>Order History</i>	37
4.7.2	<i>Managing Stock</i>	37
5	Conclusion	39
6	References	41
7	Appendix - Summary in Swedish.....	43
7.1	Bakgrund	43
7.2	Målsättning och metoder	43
7.3	Resultat	44

Figures

Figure 1. Commerce Kickstart Demo Shop.....	17
Figure 2. Drupal 7 installation GUI.....	19
Figure 3. Fields of clothing content type on Fitspire.....	22
Figure 4. Fields of clothing product variation on Fitspire.....	23
Figure 5. Clothing content type teaser display on Fitspire.....	25
Figure 6. Clothing product variation teaser display on Fitspire.....	25
Figure 7. The product view, a HTML list of nodes rendered with the teaser display....	26
Figure 8. The product view as it looks for site visitors.....	27
Figure 9. Fitspire facets.....	28
Figure 10. The configuration of an individual facet.....	29
Figure 11. Shopping cart on Fitspire.....	30
Figure 12. Shipping rule for shipping outside Denmark.....	31
Figure 13. QuickPay configuration, all sensitive information obscured.....	32
Figure 14. The order confirmation page on Fitspire.....	33
Figure 15. The rules for sending a notification message to the warehouse when an order has been completed.....	35
Figure 16. Product recommendations on Amazon.....	37

Tables

Table 1. A comparison of WordPress, Joomla and Drupal.....	14
--	----

Abbreviations and Definitions

- CMS – Content Management System
- SKU – Stock Keeping Unit
- SEO – Search Engine Optimization
- SQL – Structured Query Language
- MVC - Model–view–controller
- GPL – GNU General Public License
- API – Application Programming Interface
- UI – User Interface
- GUI – Graphical User Interface
- Drupal core – a basic installation of Drupal without any extra modules or themes
- Module – An addon for Drupal which adds functionality
- Contrib module – a module created by someone from the Drupal community and published on drupal.org
- Theme – A collection of files that produce a graphical interface according to a specific design
- Node – A piece of content in Drupal, like an article or a page

INTRODUCTION

1.1 Background and Scope

The contents of this thesis is based on experience from implementing Drupal Commerce on a Drupal page, called Fitspire, built by my employer Genero. Fitspire was commissioned by a Danish company and its purpose is to be a fitness portal, where people can visit the site to find personal trainers, classes as well as blogs and through the web shop they'll order their fitness equipment. My part of this project was implementing this web shop. This implementation was done with the help of Commerce Kickstart 2.

Drupal is a content management system (CMS), a ready-made platform upon which a developer easily can build up a website. Drupal Commerce is a collection of Drupal modules you can use to build and flesh out a web shop. It includes modules for everything from sales tax to warehouse logic and stock management. Commerce Kickstart 2 is a web shop template which mainly uses Drupal Commerce modules to build a simple web shop.

In the theoretical part of this thesis an overview of CMS, Drupal, Drupal Commerce and Commerce Kickstart is given, and an explanation as to why these particular solutions were chosen for this project and as a focus for this thesis. The practical part focuses on the different steps of implementing a web shop using Drupal Commerce.

1.2 Objective and Methods

The Objective of this thesis is to describe the process of creating a web shop using Drupal Commerce with Commerce Kickstart as a model. It will describe the different steps of creating a web shop based on practical experiences from the Fitspire project. The objective of the Fitspire project was to implement a fully functional web shop that supports a wide range of products. It had to work smoothly and look good, making customers want to return.

The theoretical part of this thesis is based on literature search and information gathering from online sources such as Drupal's own webpage and the Commerce project page.

The practical part is based on experiences gathered from the practical implementation of the Fitspire web shop.

1.3 Thesis Structure

The rest of the thesis is structured as follows. Chapter 2 is an over-view of CMSs and why Drupal was chosen. Chapter 3 is an over-view of e-commerce platforms and why Drupal Commerce was chosen. Chapter 4 describes the different steps of creating a Drupal Commerce site. Chapter 5 then concludes the thesis.

2 CONTENT MANAGEMENT SYSTEMS

A CMS (Content Management System) is a pre-built web application that makes up the base infrastructure of a site. A CMS allows users to publish, edit, organize and delete content, all from a central interface. Developers can then build their site on top of this infrastructure, adding features and styling as required, without wasting time on building a site from scratch.

CMSs are often used for blogs, web shops, news outlets and company homepages. Some of the most used CMSs are Drupal, WordPress and Joomla which all are built using PHP and MySQL. There are also CMSs built on other platforms, such as Umbraco which is written in C#.

Many CMSs also allow the users to download modules (called plugins in WordPress or extensions in Joomla), which can either extend already existing functionality or add completely new functionality to the CMS, and themes which can be used to change the appearance of the site and then further customized to suite the need of that particular site.

2.1 WordPress

WordPress is the most popular CMS in the world, where over 60% of all sites that are built using CMSs (or 23.5% of all websites total) are built using WordPress (W3Techs, 2015). Some of the notable sites that use WordPress include The New York Times, CNN and Forbes (WordPress, 2015).

WordPress was first released back in 2003 by Matt Mullenweg and Mike Little as a fork from of the b2/catalog blogging platform (Mixergy, 2009) and is licensed under the GPLv2. WordPress offers both a hosted, ad-supported version at wordpress.com and a self-hosted version at wordpress.org. The latest stable release as of April 2015 of this thesis is 4.1.1, released in February 2015. It is worth mentioning that while the current

version starts with a 4 that does not mean that there have been 4 major WordPress versions; due to the way WordPress numbers its versions the move from 3.9 to 4.0 was of no more significance than the move from 4.0 to 4.1 (WordPress.org, 2015).

WordPress is the ideal CMS for someone who wished to create a simple website or blog and due to its popularity also has the largest community and the most community created themes and modules. WordPress has a very simple backend interface that is easy to learn to use even for the less tech savvy.

WordPress has a PHP and MySQL backend and to customize how something is displayed a user needs to understand a fair bit of PHP and how to use WordPress hooks. So, while easy to use for simple purposes, mastering it isn't as easy and the more customization is done this way, the more bloated the site will be.

2.2 Joomla

Joomla is the second most popular CMS in the world and has a 7.4% market share among sites using CMSs (W3Techs, 2015). Joomla is often considered a compromise between WordPress and Drupal. It is more customizable than WordPress, but doesn't require the technical expertise Drupal does. Some of the notable sites that use Joomla include International House of Pancakes and Harvard University (Joomla, 2015).

Joomla was released in 2005 as a fork of the Mambo project and is licensed under the GPLv2. The latest release of Joomla as of April 2015 is version 3.4.1 which was released in March 2015. The Joomla version number doesn't accurately indicate the number of major releases; as an example Joomla moved from version 1.7 to 2.5 since they had decided that x.5 versions would be their Long Term Support versions (a practice they have since abandoned).

Joomla markets itself as an easy to install but highly configurable solution. It claims to specialize in easy but flexible sites with a good Search Engine Optimization, multilingual and social networking support. Joomla is popular among some coders for using MVC (Model View Control), an architectural pattern separating components in models,

views and controllers. MVC separates application logic from the user interface, leading to a more modular program.

2.3 Drupal

Drupal is the third most popular CMS with a 5.1% market share of sites that use CMSs (W3Techs, 2015). Drupal is the most complex of the top three CMSs, with the most technological expertise required to build and maintain a site, but also the most flexible.

Drupal is an open source CMS developed by a huge community (37 186 developers and over 1.1 million registered community members (Drupal, 2015)) and distributed under the GNU General Public License. It is used on over 1.1 million sites worldwide (Drupal, 2015). Drupal is built using PHP and MySQL and the first version was written by Dries Buytaert back in 2001.

Currently most sites use Drupal 7, which was released in January 2011, and the release of Drupal 8 is just around the corner. The latest version of Drupal 7, 7.36 was released in April 2015. Unlike WordPress and Joomla the version numbering of Drupal accurately reflects the number of major versions released so far.

Drupal core (a standard installation of Drupal) contains a backend for basic content creation, account creation and management and layout management. Using only the Drupal core you can already make a solid blogging platform, news site or similar. But the real strength in Drupal lies in its modular nature and a huge module creating community. Currently there are 29 620 modules and 2 105 themes available for download on Drupal's site (Drupal, 2015), giving users thousands of additional features.

These modules add additional configuration options and features to Drupal, and with such a huge variety of modules, there is a good chance that a user needs some special feature for their site, someone has already built a module to support that (or a similar) feature. The variety of themes gives the user plenty of options for picking a core theme, which can then be configured until the site looks exactly like the user imagined it.

Drupal is also considered a secure platform, secure enough for several major corporations and government organizations (such as The White House (The White House, 2015)) to use as their default website platform.

2.4 Summary

WordPress is a simple platform, which focuses on creating blogs and simple pages and that is where it excels. The admin interface and overall configurability of a Joomla site is a bit more advanced than that of WordPress, but over the last few years that gap has shrunk significantly as WordPress has improved at a steady pace. Drupal is the most complicated of the three, but also the most versatile, allowing for complex relationships between content and a very robust backend.

A good example of the differences between the CMSs is what content types are available after a fresh install:

WordPress has two; page and post, clearly illustrating its purpose as a log platform with support for some basic pages. To add more content types in WordPress the user has to know a fair bit of PHP and even then they're just extended versions of one of the two basic content types.

Joomla has managed content types by merging the two that WordPress uses into one called article. Articles can then be configured to act more as posts or pages based on the need of the user. To add more content types in Joomla modules are needed.

A fresh install of Drupal has the same content types as WordPress, named article and basic page. Where Drupal excels is that right out of the box you can edit those content types by adding new fields, removing fields, managing how the fields are displayed or add new content types if need be.

While both WordPress and Joomla support both blogs and web shops, Drupal is much more flexible and easily extended with different modules, making it ideal for more complex sites. Take the Fitzpire project as an example. Not only should it support a web

shop, it should also allow users to create their own fitness blogs, allow venues like gyms to fill in their opening hours, location and even maintain their own blog. The wide variety of features required by the Fitspire project, and the heavy integration in between those features, made Drupal the only option.

Table 1. A comparison of WordPress, Joomla and Drupal

	WordPress	Joomla	Drupal
CMS Marketshare ¹	60%	7.4%	5.1%
Release Date	27.5.2003	15.1.2001	16.9.2005
Latest Version	4.1.1 (Feb 2015)	3.4.1 (Mar 2015)	7.36 (Apr 2015)
License	GNU GPLv2+	GNU GPLv2+	GNU GPLv2+
Platform	PHP	PHP	PHP
Cost	Free	Free	Free
Free Modules ²	~37000	~4500	~30000
Free Themes ²	~1700	Unknown	~2100
Flexibility	Low	Medium	High
Difficulty	Low	Medium	High
¹ (W3Techs, 2015)			
² Source: as available on wordpress.org , joomla.org and drupal.org			

3 E-COMMERCE SOLUTIONS

As emerged from Chapter 2, Drupal is the most suitable CMS for development of more complex web sites, such as Fitspire, and therefore the rest of the thesis will focus on Drupal. This chapter will compare the two major commerce frameworks for Drupal, Drupal Commerce and Ubercart, and to be thorough also the ecommerce platform Magento.

3.1 Magento

Magento is an open source CMS focusing solely on e-commerce (Magento, 2015). Magento was first released in March 2008 and is according to a study by aheadWorks made in October of 2014 (Shatkov, 2014) the most popular e-commerce platform with a 25.6% market share. Magento is built using PHP with a MySQL database.

Magento offers users a free Community Edition and two commercial versions, Enterprise Edition and Magento Go. Enterprise Edition is derived from the Community edition, but offers additional features and functionality for an annual maintenance fee. Magento Go is a cloud-based e-commerce solution which is hosted on Magentos servers.

3.2 Ubercart

Ubercart was the first major e-commerce framework created for Drupal, released for Drupal 5 in 2008 and Drupal 6 in 2009 (Ubercart, 2015). Ubercart was released in 2012 for Drupal 7, over a year after the initial release of Drupal 7, which led to it losing some of its users to Drupal Commerce. As of April 2015 barely 48 000 Drupal sites report using the Ubercart module (Drupal, 2007).

Ubercart is a very strong solution that works well out-of-the-box. It is fairly easy to learn and intuitive to use. When it comes to quickly setting up a simple web shop in Drupal, Ubercart is definitely a strong contender.

3.3 Drupal Commerce

Drupal Commerce is a commerce framework which can be deployed on any Drupal 7 site. Drupal Commerce adds Commerce related APIs to Drupal, such as checkout, products and payment methods. (Dianiska, 2015) Drupal Commerce consists of a set of core modules to create the Commerce framework and can then be extended to support the features needed by a web shop by using contrib modules made specifically for Drupal Commerce. Contrib modules are modules released by Drupal users which are not included in Drupal core but are (or were at a time) actively supported on drupal.org.

Drupal Commerce was created as an alternative to Ubercart for Drupal 7, since at first it seemed Ubercart would not be ported from Drupal 6 to Drupal 7. It was created by a development team called the Commerce Guys and version 1.0 was released in 2011. At the time of writing this thesis over 56 000 sites are currently, as of April 2015, using Drupal Commerce. (Drupal, 2009)

3.3.1 Commerce Kickstart

Commerce Kickstart is a packaged installation profile of Drupal Commerce, containing the modules and configurations needed to set up a basic Drupal Commerce shop. It also contains some sample products, shipping and pricing rules and views that list said products (Figure 1). It's a good starting point for any web shop, or if you don't wish to use it as a core for your web shop it can still be used as a valuable example for how to configure several different areas of the shop.

In the Fitspire project it was used in the latter way, as an example, while configuring our shop from scratch. This was done to avoid unnecessary extra features offered by Kickstart, which would have added extra clutter to the web shop.

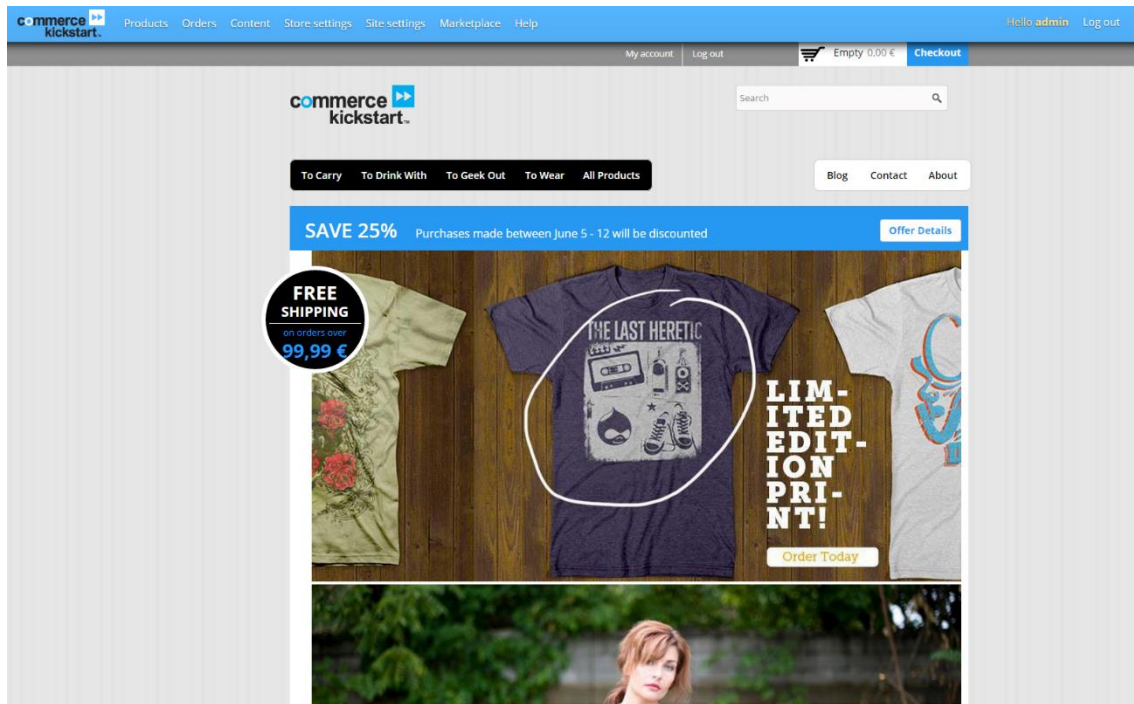


Figure 1. Commerce Kickstart Demo Shop

3.4 Summary

While Magento makes for an extremely strong e-commerce platform, it is also very specialized. If the site is to feature something besides a web shop, like a blog or a news feed another platform is needed. Beyond that the free Community Edition is fairly limited compared to the commercial Enterprise Edition which stands more on an equal footing with the free Ubercart and Drupal Commerce. The price barrier and lack of flexibility means Magento is the first platform to fall from this comparison.

Drupal Commerce versus Ubercart on the other hand is a topic which still is hotly debated among developers. Ubercart provides a solution that will work straight out of the box, while Drupal Commerce will require significantly more initial configuration. This is because Drupal Commerce installs with a much more skeletal profile and then lets the user install the modules needed for their specific site, making it a significantly more flexible solution.

While Ubercart works well as a standalone solution, Drupal Commerce integrates more heavily with Drupal and popular Drupal modules like Views and Rules, meaning it will be a lighter solution for sites already using these modules. Beyond that, Drupal Commerce was a lot quicker to release a stable version for Drupal 7 and all indications are that the same will be true for Drupal 8. This indicates that while both solutions probably will work in new versions of Drupal, Drupal Commerce seems to be the solution of choice for those who want to migrate to newer Drupal versions early.

All these factors makes Drupal Commerce the preferred e-commerce solution for a more complex project like Fitspire.

4 BUILDING A COMMERCE SITE

This chapter will deal with the different parts of setting up a Drupal Commerce site. It assumes some knowledge of how Drupal works and will be using the Fitspire project as a source of examples for how the different parts can be set up and illustrations.

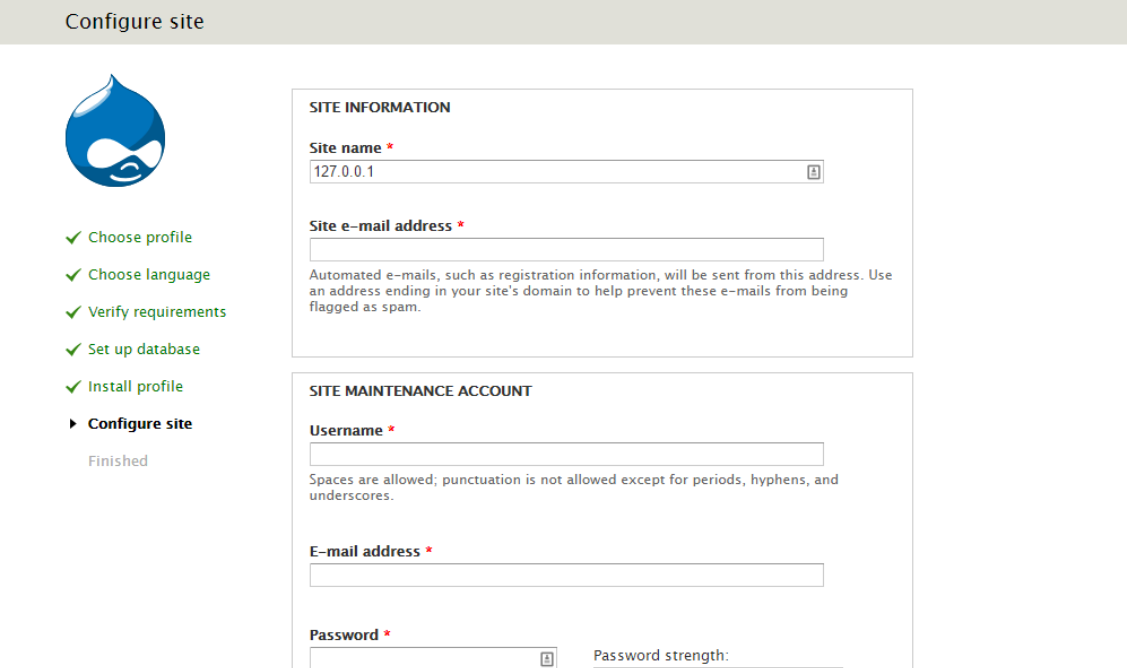
4.1 Installing Drupal Commerce

4.1.1 Drupal Core Installation

Installing Drupal is fairly simple; download it from drupal.org, unpack the package into a browser accessible directory, create an empty database for it to use and run it in your browser. Then follow the step by step instructions (Figure 2).

Alternatively you can install it using drush running the following commands:

```
drush dl drupal-7.x
drush site-install standard --account-name=admin --account-pass=admin --db-url=mysql://Your-MySQLUser:RandomPassword@localhost/YourMySQLDatabase
```



The screenshot shows the 'Configure site' step of the Drupal 7 installation process. On the left, a vertical list of steps is shown with green checkmarks: 'Choose profile', 'Choose language', 'Verify requirements', 'Set up database', 'Install profile', and 'Configure site' (which is highlighted with a right-pointing triangle). Below this list, the word 'Finished' is displayed. To the right of the list is the Drupal logo. The main area contains two form sections: 'SITE INFORMATION' and 'SITE MAINTENANCE ACCOUNT'. The 'SITE INFORMATION' section has a 'Site name *' field containing '127.0.0.1' and a 'Site e-mail address *' field. Below the email field is a note: '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.' The 'SITE MAINTENANCE ACCOUNT' section has a 'Username *' field, an 'E-mail address *' field, and a 'Password *' field. To the right of the password field is a 'Password strength:' indicator.

Figure 2. Drupal 7 installation GUI

4.1.2 Drush

Drush is a useful command line tool for installing and maintaining a Drupal installation on a server. It can be used to download, enable and disable modules, to install module dependencies (which it automatically checks when you try to enable a module) and to update Drupal and the modules installed. Some of the other useful features are the ability to clear the site cache from the command line (the first thing to do when the site starts behaving oddly) and the ability to clone a database from a live environment to the development environment (or vice versa) with a single command.

4.1.3 Setting up Drupal Commerce

This can be done in two ways. The easy way is to just download and unpack a Commerce Kickstart package and modify that to suite your web shop needs. The other way is to do a normal Drupal installation and then add the module you need for your particular web shop. The latter way was chosen on the Fitspire project, though a basic Kickstart installation was created to use as an example.

To get a standard installation of Drupal to run a working Drupal Commerce installation the following contrib modules from Commerce Kickstart were copied to it (situated in the folder `profiles\commerce_kickstart\modules\contrib` starting from the Commerce Kickstart base folder):

- *addressfield* – adds support for address fields, needed for shipping information
- *admin_menu* – **not part of Commerce Kickstart** and isn't required for Drupal Commerce to run, but makes working with Drupal much more pleasant (by adding dropdowns to the admin menu so users don't have to navigate through several pages but instead can pick the site they want through the dropdowns)
- *commerce* – the core module of Drupal Commerce
- *commerce_autosku* – lets product variations automatically generate a SKU, saves time
- *commerce_backoffice* – creates clearer pages that list products and orders
- *commerce_checkout_progress* – shows the customer which step they're currently on in the checkout process

- *commerce_discount* – support for discounts
- *commerce_extra_price_formatters* – supports different ways of printing the price
- *commerce_flat_rate* – support for flat rate shipping
- *commerce_message* – send message to customer at different stages of the order process
- *commerce_paypal* – adds paypal support
- *commerce_repair* – **not part of Commerce Kickstart**, repairs some issues that might come up when installing Drupal Commerce this way
- *commerce_search_api* – support for searching through products and better filters
- *commerce_shipping* – support for shipping
- *ctools* – required by several commerce elements
- *date* – required by backoffice
- *ds* – Display Suite, **not part of Commerce Kickstart**, but makes styling content types much easier
- *entity* – required by commerce core
- *entityreference* – required by commerce message, among others
- *eva* – required by backoffice
- *facetapi* – better filters for search
- *inline_conditions* – required by commerce discount
- *inline_entity_form* – required to create product variation to node links
- *message* – required by commerce message
- *message_notify* – required by commerce message
- *rules* – required by commerce core
- *search_api* – required by commerce_search_api
- *search_api_db* – adds extra features to search_api
- *search_api_ranges* – adds extra features to facetapi
- *search_api_sorts* – required by facetapi
- *views* – required by several core commerce modules
- *views_bulk_operations* – required by backoffice
- *views_megarow* – required by backoffice

4.2 Creating the Products

4.2.1 Nodes

Nodes are the building blocks for content in Drupal. A node is a piece of content with fields according to its content type. The basic content types in Drupal core are basic pages and articles. While they both share title and body fields and article additionally support tags and an article image.

When creating a web shop the first decision when it comes to nodes is whether to squeeze all products into one content type or wish to create several content types which are more tailored to the type of product. Fitspire went with the latter option.

Fitspire launched with three content types; training equipment (called training), nutritional products (called nutrition) and clothing (Figure 3). Training and clothing haven't really got any unique fields of their own in the content type, but nutrition supports extra fields such as nutritional facts and ingredients on this level. Most of the product specific fields are stored in the product variation; the content type only stores such fields which are common to all the variations of that product, like name, description and product type.

Label	Machine name	Field type
+ Title	title	Node module element
+ Sprog	language	Language selection
+ Body	body	Long text and summary
+ Product Type	field_product_type	Term reference
+ Product variations	field_product	Product reference
+ Product Status	field_product_status	Term reference
+ Product Discount	field_product_discount	Term reference
+ URL path settings	path	Path module form elements
+ URL redirects	redirect	Redirect module form elements
+ Meta tags	metatags	Meta tag module form elements.

Figure 3. Fields of clothing content type on Fitspire

4.2.2 Product Variations

Product variations are the real workhorses when it comes to storing product. While the node stores the basic info, the product variation stores the more specific info such as product SKU (Stock Keeping Unit, a unique identifier for each distinct product), price, images and stock. Beyond this is also where product variation specific data is stored; such as size and colour for clothing and flavour for nutrition products.

This allows the shop to sell the same shirt, but in different colours and sizes (and possible at different prices depending on colour and size) while only creating one base product (Figure 4). By attaching images to the product variation rather than the product node the image will also change based on the colour of the shirt you pick, so you get an idea how that shirt will look.

Creating a new product variation is fairly simple. Just go to Store, Configuration, Product Variation Types and Add Product Variation Type. When creating a Product Variation, check that “Create matching product display type” is checked. “Automatically Generate SKU” is also a useful option that should be checked, using a pattern such as “[commerce-product:title]-[commerce-product:product-id]”.

Label	Machine name	Field type
+ Product SKU	sku	Product module SKU form element
+ Titel	title	Product module title form element
+ Price	commerce_price	Pris
+ Images	field_images	Billede
+ Color	field_color	Term reference
+ Size	field_size	Term reference
+ URL redirects	redirect	Redirect module form elements
+ Status	status	Product module status form element
+ Stock	commerce_stock	Decimal
+ Disable stock for this product	commerce_stock_override	Boolean

Figure 4. Fields of clothing product variation on Fitspire

4.3 Displaying the Products

4.3.1 Display Suite

Display Suite is a module which allows the user to manage how nodes are displayed. Each node can have several displays, like full content and teaser. Display Suite then allows you to select which fields are printed out in which display and how they are printed. It also allows the user to add classes to the different fields or the display as a whole and to group fields for easier styling.

Consider how a product should be displayed. The teaser is fairly simple; an image on top, then the product name, price and finally a “add to cart” button. Fitspire displays two additional fields in the teaser; how many product reviews have been written for a product and what the average five star rating is for this product and how many professionals have recommended the product.

In full content Fitspire utilizes yet another Display Suite feature; layouts. Layouts are predefined ways of grouping your content. Fitspire’s full content display for products used a layout called 2 columns stacked. What this means is a layout that has a full width column on top (called Top), then two equal width columns under it (called Left and Right) and another full width column at the bottom (called Bottom). In Top Fitspire placed the title, then the product images in Left, the “add to cart” form, price and star rating in Right and the description and reviews in Bottom.

When dealing with Drupal Commerce’s product variations it’s worth noting that to style a field that is defined in product variations it can be positioned in the content type display (Figure 5), but any other settings for how that field is printed needs to be set in the product variations display (Figure 6) with the same name. So if a user wishes to change the image style of the image printed in the product teaser he has to go to the product variation teaser and change the display setting for the image style there, then position the image in the product content type.

Field	Region	Label	Format
Content			
+ Product: Images	Content		Visible <small>Modify the settings for this field on t</small>
+ Product Status	Content	<Hidden>	Rendered taxonomy term
+ Product Discount	Content	<Hidden>	Rendered taxonomy term
+ Title	Content	<Hidden>	Default
+ Product: Price	Content		Visible <small>Modify the settings for this field on t</small>
+ Vote	Content	<Hidden>	As Stars
+ Comment count	Content	<Hidden>	Default
+ Professional recommendations	Content	<Hidden>	Default
+ Read more	Content	<Hidden>	Default

Figure 5. Clothing content type teaser display on Fitspire

Field	Region	Label	Format
Content			
+ Title	Content		Visible
+ Images	Content	<Hidden>	Image 1
+ Price	Content	<Hidden>	Formatted amount
+ Color	Content	<Hidden>	Rendered taxonomy term
+ Size	Content	<Hidden>	Link
+ Status	Content		Visible
+ Flag: Favourite product	Content		Visible

Figure 6. Clothing product variation teaser display on Fitspire

4.3.2 Views

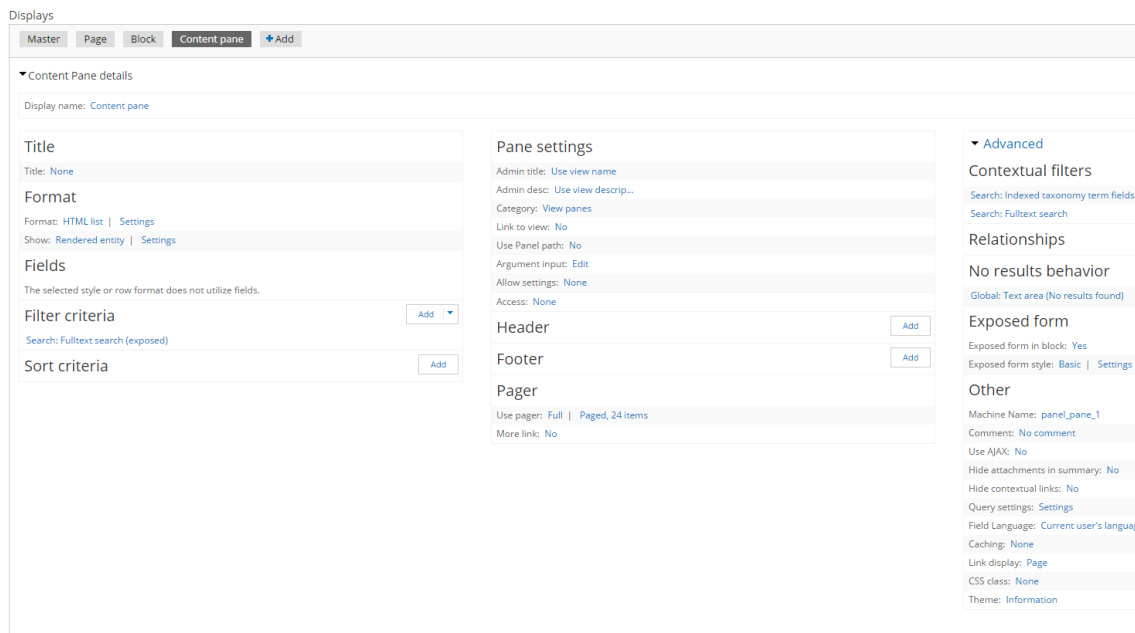
Views is one of the most popular Drupal 7 modules and is an extremely complex and flexible module. Trying to explain views in its entirety would require a whole thesis of its own, this chapter will only briefly touch its surface.

A simple way to explain views is to say that it's a module which creates lists of content on a Drupal site. When listing content views can either show a display (like the ones created in Display Suite) or it can print one or more fields.

The thing that makes views really useful are the filters. There are the normal, “static” filters which can only be changed by editing the view. An example of this is filtering by content type. Then there’s the contextual filter which filters based on the URL, the user or the current language. And finally there are exposed filters which allows a user to select the value to filter for in the front end.

Views can further be sorted using the different fields; either statically when configuring a view from the backend or through exposed sorting options that the frontend users can apply. The views module also has built in support for paging, so a view can be set either to show all the content on one page or a set number of items per page using paging (or just a set number of items without paging, if that’s what is needed).

On Fitspire the shop’s frontpage has a view which shows the 3 most popular products, based on how many have been purchased in total. Then the taxonomy term pages for the different product types all use the same view that fetches all products, but applies a contextual filter based on which product type the page represents (Figure 7 & 8). To further narrow down the product search there is a collection of facet filters.



The screenshot shows the 'Displays' configuration page for a view named 'Content pane'. The interface is divided into several sections:

- Content Pane details:** Shows the display name as 'Content pane'.
- Title:** Set to 'None'.
- Format:** Set to 'HTML list'.
- Fields:** A note states 'The selected style or row format does not utilize fields.'
- Filter criteria:** Includes 'Search: Fulltext search (exposed)'.
- Sort criteria:** Includes 'Search: Fulltext search (exposed)'.
- Pane settings:** Includes fields for 'Admin title', 'Admin desc', 'Category', 'Link to view', 'Use Panel path', 'Argument input', 'Allow settings', and 'Access'.
- Header:** Includes an 'Add' button.
- Footer:** Includes an 'Add' button.
- Pager:** Set to 'Full' with 'Paged, 24 items' and 'More link: No'.
- Advanced:** Includes sections for 'Contextual filters', 'Relationships', 'No results behavior', 'Exposed form', and 'Other'.

Figure 7. The product view, a HTML list of nodes rendered with the teaser display

ERNÆRING ▾

Protein og tilskud

- Proteinpulver
- Weight Gainer
- Kreatin
- Amino acids
- Vitaminer og mineraler
- Andre tilskud
- Energi & Protein Bars
- Driks
- Shake 'n' bake
- Fødevarer

ACCESSORIES ◀

GAVEKORT

Search by

Title (A-Z) ▾

Price

14€ ————— 399€

Flavor

- Strawberry (11)
- Vanille (10)
- Chocolate (9)
- Neutral (9)
- Lemon (5)
- Blueberry (3)
- Lakrids (3)
- Pistacie (3)
- Banana (2)
- Orange (2)

Show more


Special diet

- Gluten free (2)
- Lactose free (2)

PROTEIN OG TILSKUD

Proteinpulver og andre tilskud er et godt supplement, som hjælper med at styrke dine muskler. I Fitspires webshop kan du finde en bred vifte af produkter, som er anbefalet af vores professionelle fagfolk inden for træning og sundhed (Fitspirators).


Vi har et bredt udvalg af forskellige typer produkter og mange smagsvarianter, så du kan finde lige det, der passer dig. Alle vore produkter er godkendt af Sundhedsstyrelsen i Danmark.



AMINO LIQUID - BODYLAB

~~139,00 DKK~~ **111,20 DKK**


[READ MORE](#)



ANABOLIC AMINO TABS

~~229,00 DKK~~ **183,20 DKK**


[READ MORE](#)




BCAA INSTANT - BODYLAB

~~229,00 DKK~~ **183,20 DKK**

[READ MORE](#)


 1 professionals recommend



BCAA PULVER - BODYLAB

~~229,00 DKK~~ **183,20 DKK**


[READ MORE](#)



BCAA TABLETTER - BODYLAB

~~169,00 DKK~~ **135,20 DKK**

[READ MORE](#)



BETA ALANINE

~~139,00 DKK~~ **111,20 DKK**

[READ MORE](#)

Figure 8. The product view as it looks for site visitors

4.3.3 Facets

Facets are a type of dynamic search filters (see Figure 9). They allow the user to narrow their search using different taxonomy terms attached to products and as each new filter is selected the results update and the facet filters update to only show the filters that are relevant to the remaining products.

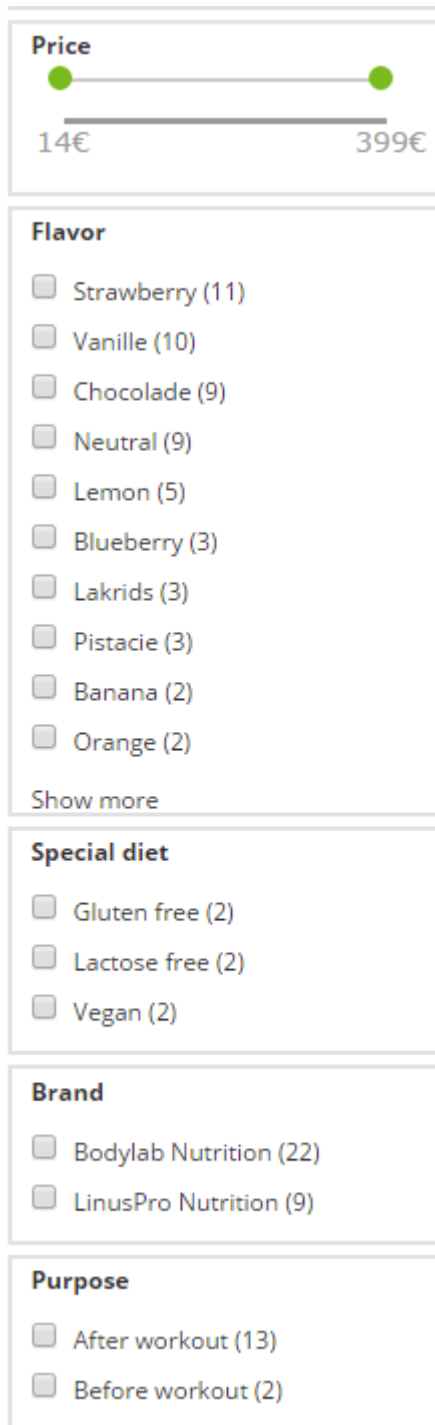


Figure 9. Fitspire facets

The facets are an extension to the Search API, a service which indexes all nodes and makes searching them quicker. The Search API can either interact with a backend indexing service such as SOLR or just create an index in the MySQL database.

To create facets, an index must be created and the fields that should be used as filters selected. Then the facet for that field should be activated and configured (Figure 10) and finally added to the page where the product list is displayed.

On Fitspire the preferred type of facet displayed as links with checkboxes, showing the number of results for that particular filter. If a facet had no matches, it was hidden. Thus a full list of all facets could be used for all content types, since those not relevant to the current content types would be automatically hidden away.

The image shows the configuration interface for an individual facet in Drupal Commerce. It is divided into two main sections: 'Display settings' and 'Global settings'.

Display settings:

- Display widget:** A dropdown menu set to 'Links with checkboxes'.
- Display the number of matching items:** A checkbox that is checked. A note below states: 'If checked, the number of matching items will be output with the color square.'
- Soft limit:** A text input field containing the number '20'. A note below states: 'Limits the number of displayed facets via JavaScript.'
- Prevent crawlers from following facet links:** A checked checkbox. A note below states: 'Add the meta="nofollow" attribute to facet links to maximize SEO by preventing crawlers from indexing duplicate content and getting stuck in loops.'
- Expand hierarchy:** An unchecked checkbox. A note below states: 'Show the entire tree regardless of whether the parent items are active.'
- Empty facet behavior:** A dropdown menu set to 'Do not display facet'. A note below states: 'The action to take when a facet has no items.'

Global settings:

The configuration options below apply to this facet across all realms.

- Operator:** Radio buttons for 'AND' (selected) and 'OR'. A note below states: 'AND filters are exclusive and narrow the result set. OR filters are inclusive and widen the result set.'
- Exclude:** An unchecked checkbox. A note below states: 'Make the search exclude selected facets, instead of restricting it to them.'
- Hard limit:** A text input field containing '50'. A note below states: 'Display no more than this number of facet items.'
- Minimum facet count:** A text input field containing '1'. A note below states: 'Only display facets that are matched in at least this many documents.'
- Flatten hierarchy:** Radio buttons for 'No' (selected) and 'Yes'. A note below states: 'Do not process hierarchical relationships and display facet items as a flat list.'
- Add facet for missing values:** Radio buttons for 'No' (selected) and 'Yes'. A note below states: 'Adds an extra facet matching documents with no value at all for this field.'

Sort and Order table:

Sort	Order
<input checked="" type="checkbox"/> Facet active Sort by whether the facet is active or not.	Descending
<input checked="" type="checkbox"/> Count Sort by the facet count.	Descending
<input checked="" type="checkbox"/> Display value Sort by the value displayed to the user.	Ascending
<input type="checkbox"/> Indexed value Sort by the raw value stored in the index.	Ascending

Figure 10. The configuration of an individual facet

4.4 The Checkout Process

The checkout process works pretty well out of the box on any Commerce installation, from the shopping cart (Figure 11) to the order review page. The only two elements that need to be configured is shipping costs (chapter 4.5) and payment system (chapter 4.6). Drupal Commerce allows the user to change the amount of pages that are part of the checkout process and which elements, such as forms for shipping and billing information, are to be shown on which page. But in the end, this feature is so well implemented that unless the design for the page is extremely specific on how the checkout should work, there isn't much that needs to be done here.




Produkt	Pris	Antal	Fjern	I alt
 Quest bars 12 stk (Banana Nut) Smagsvariant: Banana Nut Gå til produkt	249,00 DKK	<input type="text" value="1"/>	Fjern	249,00 DKK
				Total: 249,00 DKK
Opdater kurven				TIL KASSEN

Figure 11. Shopping cart on Fitspire

4.4.1 Shipping Costs

Before customers can start ordering products from the web shop, they need to be given some shipping options. Drupal Commerce has a built in shipping UI where the available shipping methods and pricing can be easily configured. The shipping interface allows the user to create rules where price varies based on such things as the shipping destination or the total value of the order (Figure 12).

Conditions

Elements

⊕ NOT Order address component comparison

Parameter: *Ordre*: [commerce_order], *Adresse*: Adresse, *Address component*: Land, *Value*: DK

+ Add condition + Add or + Add and

Actions

Elements

⊕ Add a rate for a shipping service to an order

Parameter: *Fragt*: Shipping outside Denmark ..., *Ordre*: [commerce-order]

+ Add action + Add loop

Figure 12. Shipping rule for shipping outside Denmark

4.4.2 Payment Systems

Payment methods in web shops can be a huge security risk. Storing credit card data and the like makes a site a very tempting target for attackers. Fortunately, Drupal Commerce gives the user the opportunity to outsource the payment system to a more reliable and experienced service, like PayPal or Amazon.

This thesis will not go into an in-depth comparison of the different payment systems, since from a Drupal Commerce standpoint they are all easily implemented by just enabling their respective module. The real concern when choosing one is what sort of cut from any transaction the service takes and what monthly fees are involved, meaning this is more a choice for an accountant than a developer.

For Fitspire the client chose to go with the payment system QuickPay, a service which accepts all major credit cards and a few local Danish ones and also integrates support for some other services such as PayPal and offers these to the customer based on how the QuickPay module is configured (Figure 13). When a customer is checking out and have reviewed their order (Figure 14) they are redirected to the QuickPay page where

they handle the payment. After that they are redirected back to the web shop and QuickPay sends the shop a confirmation that a payment has been made.

Order

Data selector *

commerce-order

<div class="rules-dsac-item rule-dsac-group" title="Site-wide settings and other global information">site... (Site information)</div>
<div class="rules-dsac-item" title="">commerce-order (Order)</div>
<div class="rules-dsac-item rule-dsac-group" title="">commerce-order... (Order)</div>

▸ Data selectors

Switch to the direct input mode

Payment settings

Merchant number *

Merchant id as shown in the QuickPay admin. NOTE: not the same as PBS id

MD5 secret *

MD5 secret as shown in the Quickpay admin.

Order id prefix

Prefix for order ids. Order ids must be unique when sent to QuickPay, use this to resolve clashes.

Sprog

Language of the user

The language for the credit card form.

Accepted payment methods

Creditcard
 3D-Secure Creditcard
 Selected payment methods

Which payment methods to accept. NOTE: Some require special agreements.

Selected payment methods

<input checked="" type="checkbox"/> Dankort <input checked="" type="checkbox"/> eDankort <input checked="" type="checkbox"/> Visa <input checked="" type="checkbox"/> Visa, issued in Denmark <input checked="" type="checkbox"/> Visa, using 3D-Secure <input checked="" type="checkbox"/> Visa, issued in Denmark, using 3D-Secure <input checked="" type="checkbox"/> Visa Electron <input checked="" type="checkbox"/> Visa Electron, issued in Denmark <input checked="" type="checkbox"/> Visa Electron, using 3D-Secure <input checked="" type="checkbox"/> Visa Electron, issued in Denmark, using 3D-Secure <input checked="" type="checkbox"/> Mastercard <input checked="" type="checkbox"/> Split payments <small>Allows for capturing payments in parts.</small> <input type="checkbox"/> Test mode <small>When active, transactions will be run in test mode, even if the QuickPay account is in production mode. Order ids will get a T appended.</small> <input checked="" type="checkbox"/> Debug log <small>Log all request and responses to QuickPay in watchdog.</small>	<input checked="" type="checkbox"/> Mastercard, issued in Denmark <input checked="" type="checkbox"/> Mastercard debit card, issued in Denmark <input checked="" type="checkbox"/> Mastercard, using 3D-Secure <input checked="" type="checkbox"/> Mastercard, issued in Denmark, using 3D-Secure <input checked="" type="checkbox"/> Mastercard debit, issued in Denmark, using 3D-Secure <input type="checkbox"/> Maestro <input type="checkbox"/> Maestro, issued in Denmark <input checked="" type="checkbox"/> JCB <input checked="" type="checkbox"/> JCB, using 3D-Secure <input type="checkbox"/> Diners <input type="checkbox"/> Diners, issued in Denmark	<input checked="" type="checkbox"/> American Express <input checked="" type="checkbox"/> American Express, issued in Denmark <input type="checkbox"/> Danske Netbetaling <input type="checkbox"/> Nordea Netbetaling <input type="checkbox"/> Forbrugsforeningen <input type="checkbox"/> Ikano <input type="checkbox"/> PayPal <input type="checkbox"/> Sofort <input type="checkbox"/> ViaBill
--	---	--

Gem

Figure 13. QuickPay configuration, all sensitive information obscured.



INDKØBSKURV

EVENT REGISTRERING

TIL KASSEN

FRAGT


GENNEMSE ORDRE

BETALING

ORDRE GENNEMFØRT

Gennemse venligst din ordre før du fortsætter

INDHOLD AF INDKØBSKURV

Produkt	Pris	Antal	I alt
 Quest bars 12 stk (Banana Nut) Smagsvariant: Banana Nut	249,00 DKK	1	249,00 DKK
	Subtotal		199,20 DKK
	Moms 25%		49,80 DKK
	Fragt - Post Danmark		0,00 DKK
	Total		249,00 DKK

KONTO INFORMATION

Brugernavn

admin

E-mail adresse

public@oxy.fi

FRAGT INFORMATION

Test
Test
12345 Test
Finland

FAKTURERINGSOPLYSNINGER

Test
Test
12345 Test
Finland
Telefon nummer:
123

HANDELSBETINGELSER

[Handelsbetingelser](#) Jeg accepterer at have læst og forstået handelsbetingelserne *

BETALING

 QuickPay Payment Gateway

NÆSTE TRIN

or [TILBAGE](#)

Figure 14. The order confirmation page on Fitspire

4.5 Order Notifications

4.5.1 Rules

Rules is one of the most popular Drupal modules, used on over 270 000 sites (Drupal, 2007). It allows users to create automated events on their sites. A rule has three parts; the event that triggers it, the conditions that need to be met for it to trigger and the action that take place if the conditions are met.

One example of this is shipping rules. The event would be when an order proceeds to the page that offers shipping options, a condition could be the order total sum and the action would be making free shipping available if the order total exceeds 200€. Another example would be notifications; the events is when the status of an order changes to a particular state, in this case a condition is unnecessary and the action is creating a mail and sending it to the customer (Figure 15).

Events

Event

Completing the checkout process

+ Add event

Conditions

Elements

None

+ Add condition + Add or + Add and

Actions

Elements

- + Create a new entity
Parameter: *Entity type*: Message, *Type*: Commerce Order: notify..., *Bruger*: [site:current-user]
Provides variables: Created entity (entity_created)
- + Set a data value
Parameter: *Data*: [entity-created:message..., *Value*: [commerce-order]
- + Save entity
Parameter: *Entity*: [entity-created], *Force saving immediately*: true
- + Send Message with Message notify
Parameter: *Message*: [entity-created], *Save on fail*: false, *Save on success*: false, *The recipient of the email*:

+ Add action + Add loop

Figure 15. The rules for sending a notification message to the warehouse when an order has been completed

4.5.2 Notifications

By default Drupal Commerce will send a customer a mail when an order has been placed. Beyond that Fitspire added mails to notify the shop manager and the warehouse when an order has been placed and a mail to customers when the order status changes to shipped.

4.6 Product Recommendations

When a customer comes to a web shop they usually have a specific product in mind that they will order. But when visiting a web shop like Amazon (Amazon, 2015), that product page will not only show the product, but also sections like “frequently bought together” and “customers who bought this item also bought” (Figure 16). At the bottom of the page, for frequent amazon customer, there will also be personalized recommendations based on previously viewed and bought products.

This is a trick used not only in web shop, but also in physical stores by placing products in such a way that the customer will be tempted to purchase things they originally didn't intend to. A good recommendation system can get customers to make a lot more purchases than they ever intended to make and therefore having a good implementation of this is essential to any web store.

Unfortunately, this is one area where Drupal Commerce is lacking. There isn't any good, free module for product recommendations. There are a few alternatives, like listing the most viewed products or the most bought products, but none of these are simple to implement.

This is why Fitspire chose to utilize a commercial product recommendation service called Nosto (Nosto, 2015). It requires the site to implement a few blocks which then track customer activity and then prints its own blocks on the site with product recommendations. Nosto imports data such as product images, titles and price and then lets the user to style them to fit well in their shop.

Frequently Bought Together



Price For All Three: **£68.12**

Add all three to Basket

Show availability and delivery details

- This item:** Drupal 7 Explained: Your Step-by-step Guide by Stephen Burge Paperback £24.99
- Drupal For Dummies by Lynn Beighley Paperback £17.94
- Drupal 7 Cookbook by James Dylan Spencer Paperback £25.19

Customers Who Bought This Item Also Bought

Product	Author	Price	Rating
Drupal For Dummies	Lynn Beighley	£17.94	4.5 stars (12 reviews)
The Definitive Guide to Drupal 7	Benjamin Melancon	£39.49	4.5 stars (6 reviews)
Drupal 7 Cookbook	James Dylan Spencer	£25.19	4.5 stars (2 reviews)
Drupal 7 Module Development	Matt Butcher	£27.99	4.5 stars (1 review)
Sams Teach Yourself Drupal in 24 Hours...	Jesse Feiler	£25.99	4.5 stars (6 reviews)

Figure 16. Product recommendations on Amazon

4.7 Store Management

Drupal Commerce is fairly easy to use for a store manager; products are added like any other content, orders are easily viewed through the order history viewed and payments are taken care of by external services. The one negative thing that can be said about Drupal Commerce is that while it's fairly easy to use, the support for bigger shops is a bit lacking, with no easy ways to enter products in bulk or similar functionality.

4.7.1 Order History

Drupal Commerce implements both an order history page for shop managers, where they can see all the orders that have been placed, and an order history page for customers where they can check up on orders they've placed.

4.7.2 Managing Stock

Drupal Commerce with the Stock module supports having products both set with a specific stock, that will be unavailable for purchase once the stock reaches zero, and products with no stock, useful for products that are only created once they've been ordered.

This system works well, though unfortunately there isn't a separate stock management UI where products stocks can be updated in bulk, rather the user has to edit each product individually which can become a bit cumbersome for major sites.

5 CONCLUSION

This thesis has provided an overview of and a critical comparison between content management systems (CMS), such as Drupal, Joomla, and WordPress. WordPress is the most popular because of its simplicity. Joomla has a steady following for providing a platform that is more flexible than WordPress, yet less complex than Drupal. However, Drupal has turned out to be the most suitable one for large sites due to its superior flexibility.

A similar comparison was made for e-commerce solutions. Magento was found to be a very good platform for sites aiming to focus solely on the web shop aspect of e-commerce. Ubercart is an easily set up solution for Drupal, but Drupal Commerce was found more suitable for a larger e-commerce site due to its superior flexibility.

The objective of this thesis was to describe the process of creating a web shop using Drupal Commerce and using Commerce Kickstart as a model for it. The text is based on literature surveys and own experience gathered from building a web shop for the fitness company, Fitspire project.

The actual implementation of Drupal Commerce, using Commerce Kickstart as a model is fairly simple. As is the case with Drupal in general, the learning curve for Drupal Commerce may be a bit tougher than some of the other solutions available, but once mastered it is a highly configurable solution that allows users to create exactly the e-commerce site they have envisioned.

The only areas where Drupal Commerce was lacking were product recommendations and bulk actions. While Nosto offers a commercial solution for product recommendations, there aren't any modules available that offer even a simple product recommendation solution. The complete lack of an interface for managing stock in bulk or adding products in bulk can also be considered a weakness in the framework that has as of yet to be addressed.

The objectives of this thesis were well met. Drupal has been identified and motivated as the most suitable CMS for a more advanced web shop, and the process of building a web shop with Drupal Commerce has been described. The research and reflections of this thesis will be beneficial as a guide to implementing Drupal Commerce both for other employees at my employer, Genero, and for anyone else new to Drupal Commerce.

6 REFERENCES

- Amazon. (2015). *Amazon*. Retrieved 4 20, 2015, from Amazon:
<http://www.amazon.co.uk/>
- Dianiska, B. (2015). *Introduction to Drupal Commerce*. Retrieved 2 14, 2015, from Acquia: <https://www.acquia.com/blog/introduction-drupal-commerce>
- Drupal. (2007). *drupal.org*. Retrieved 4 20, 2015, from Ubercart:
<https://www.drupal.org/project/ubercart>
- Drupal. (2007). *Rules*. Retrieved 4 19, 2015, from drupal.org:
<https://www.drupal.org/project/rules>
- Drupal. (2009). *Drupal Commerce*. Retrieved 4 17, 2015, from Drupal.org:
<https://www.drupal.org/project/commerce>
- Drupal. (2015). *Frontpage*. Retrieved 2 11, 2015, from Drupal.org:
<https://www.drupal.org/>
- Drupal. (2015). *Usage statistics for Drupal core*. Retrieved 2 11, 2015, from Drupal.org: <https://www.drupal.org/project/usage/drupal>
- Joomla. (2015). *Joomla*. Retrieved 3 22, 2015, from What is Joomla?:
<http://www.joomla.org/about-joomla.html>
- Magento. (2015). *Magento*. Retrieved 4 21, 2015, from Magento: <http://magento.com/>
- Mixergy. (2009). *The Biography Of WordPress – With Matt Mullenweg*. Retrieved 4 16, 2015, from Mixergy: <http://mixergy.com/interviews/the-biography-of-wordpress-with-matt-mullenweg/>
- Nosto. (2015). *Nosto*. Retrieved 4 20, 2015, from Nosto: <http://www.nosto.com/>
- Shatkov, D. (2014). *E-commerce Platforms Popularity Study, October 2014*. Retrieved 4 20, 2015, from aheadWorks: <http://blog.aheadworks.com/2014/10/e-commerce-platforms-share-investigation-october-2014>

The White House. (2015). *Developers*. Retrieved 2 11, 2015, from The White House:
<http://www.whitehouse.gov/developers>

Ubercart. (2015). *Ubercart*. Retrieved 4 21, 2015, from Ubercart:
<http://www.ubercart.org/>

W3Techs. (2015). *W3Techs - Web Technology Surverys*. Retrieved 3 22, 2015, from
Usage of content management systems for websites:
http://w3techs.com/technologies/overview/content_management/all

WordPress. (2015). *WordPress.com*. Retrieved 3 22, 2015, from Notable WordPress
Users: <https://wordpress.com/notable-users/>

WordPress.org. (2015). *Version Numbering*. Retrieved 4 16, 2015, from WordPress.org:
<https://make.wordpress.org/core/handbook/how-the-release-cycle-works/version-numbering/>

7 APPENDIX - SUMMARY IN SWEDISH

7.1 Bakgrund

Innehållet i detta examensarbete baserar sig på min erfarenhet av att implementera Drupal Commerce på en Drupal-sida, kallad Fitspire, som har byggts av min uppdragsgivare, Genero. Fitspire beställdes av ett danskt bolag, och dess syfte är att vara en fitness-portal, där människor kan hitta personliga tränare, träningsklasser, fitness-bloggar och en webbshop, där de kan beställa gymutrustning. Min del av projektet var att implementera webbshopen, med hjälp av Commerce Kickstart. Fitspire-projektets målsättning var att implementera en välfungerande webbshop som möjliggör ett stort utbud av olika produkter.

Drupal är ett innehållshanteringssystem, en färdig plattform på vilken man relativt lätt kan bygga en webbsida. Drupal Commerce är en samling av Drupal-moduler som kan användas för att skapa en webbshop och bygga ut den. Det finns moduler för allt från att beräkna skatter till varuhuslogik och lagerhantering. Commerce Kickstart är en webbshopmodell som främst använder sig av Drupal Commerce för att bygga en enkel webbshop.

7.2 Målsättning och metoder

Examensarbetets målsättning är att beskriva implementeringsprocessen för en webbshop med hjälp av Drupal Commerce, genom att använda Commerce Kickstart som modell.

I den teoretiska delen av arbetet ges en allmän överblick över innehållshanteringssystem, som Drupal, Drupal Commerce och Commerce Kickstart, och en förklaring till varför just dessa lösningar valdes för projektet och som tema för examensarbetet. Den praktiska delen fokuserar på vad allt som måste tas i beaktande när man implementerar en webbshop med hjälp av Drupal Commerce.

Den teoretiska delen av detta examensarbete är baserat på litteratursökning och insamling av information från nätkällor, så som Drupals egna hemsida och Commerce

projektets hemsida. Den praktiska delen baserar sig på erfarenhet från implementeringen av Fitspire projektets webbshop.

7.3 Resultat

I examensarbetet ges först en översikt över och en kritisk jämförelse av olika innehållshanteringssystem, så som Drupal, Joomla och WordPress, och motiveras varför Drupal valdes. WordPress är det populäraste innehållshanteringssystemet, eftersom det är väldigt lätt att lära sig använda. Joomla har sina användare bland dem som vill ha en plattform som är mera flexibel än WordPress, men enklare än Drupal. Drupal valdes som den lämpligaste plattformen för en stor webbshop på grund av sin överlägsna flexibilitet jämfört med de andra.

En liknande översikt gavs över olika webbshopsplattformar som också jämfördes med varandra. Magento visade sig vara en väldigt bra plattform för enkla webbshoppar, men för enkel för sidor som kräver funktionalitet utöver en normal webbshop. Ubercart är en enkelt implementerad webbshopslösning för Drupal, men Drupal Commerce ansågs vara lämpligare för stora webbshoppar, eftersom den, som plattform är klart flexiblere.

Drupal Commerce är väl integrerat med Drupal och använder sig av flera populära moduler så som Views och Rules för sin funktionalitet.

Själva implementeringen av Drupal Kickstart, med användning av Commerce Kickstart som modell, är rätt så enkel. Att lära sig använda Drupal Commerce kan vara mera utmanande (eller Drupal över lag) än att lära sig andra webbshopsplattformar, men när man väl lärt sig det så är Drupal Commerce ett väldigt konfigurerbart system som låter användarna skapa exakt den webbshop de föreställt sig.

Speciellt nämnvärd bland Drupal Commerce funktioner är sättet den delar produkter in i så kallade displays och variations. Displays innehåller allmänna detaljer, så som pro-

duktens namn och beskrivning. Variations innehåller, så som namnet antyder, variationer av produkten, så som färg, storlek och pris. Detta leder till en väldigt klar och tydlig struktur.

De enda områdena där Drupal Commerce var bristfällig var produktrekommendationer och massåtgärder. Medan Nosto erbjuder en kommersiell lösning för produktrekommendationer finns det för tillfället inga moduler tillgängliga som skulle erbjuda ett enkelt, kostnadsfritt alternativ. Den totala bristen på ett användargränssnitt för att hantera lagernivåer eller skapandet av produkter som massåtgärder kan även anses vara en svaghet i plattformen som ännu inte åtgärdats.

Examensarbetets målsättning nåddes. Valet av Drupal som det lämpligaste innehållshanteringssystemet för byggandet av en webbshop motiverades, och processen att bygga en webbshop med hjälp av Drupal Commerce beskrevs. Forskningen och reflektionerna i examensarbetet kommer att vara till nytta som en guide då Drupal Commerce implementeras av nya användare.