



AUSTRALIA'S BIGGEST AND BEST RANGE OF LASER CUT DIY SUPPLIES

Crafty Cuts Laser

System Documentation

Version 1.1

Jack, Fiona, Nasser
2-15-2016



Introduction

This document is intended to be used as a basic introductory guide to the system, to aid any future developers in maintaining the system and or updating it at a later point in time. The most complex if any

Hosting Service
















The host for the website is hoopla hosting. They are a company based out of New Zealand and were selected primarily for its support of CakePHP and high level of customer support.

Framework

The framework used for this system is CakePHP. Cake PHP is a free to use rapid deployment framework which is intended to allow developers to rapidly develop websites and systems. This particular version of cake is 3.X. More about CakePHP can be read here: <http://cakephp.org/>

The System source code

The source code for the system is located in the SRC folder, from this folder there are two folders of particular interest. The controller folder and the template folder. The vast majority of information and any changes that would need to be made to the system can be done in these two folders.

	Name
	bin
	config
	logs
	plugins
	src
	tests
	tmp
	vendor
	webroot
	composer.json
	composer.lock
	index.php
	phpunit.xml.dist
	README.md
	src.zip



AUSTRALIA'S BIGGEST AND BEST RANGE OF LASER CUT DIY SUPPLIES

The template folder contains code which controls how a page looks and is interacted with, the controller controls the flow of information among other things. The way the folders within their respective folders are labeled are fairly clear.

Controllers

The controllers are labeled as shown below. These have to do with key functions of the system, materials controller handles materials, FinishesController finishes etc.

	Name
	Component
	AppController.php
	CategoriesController.php
	ColorsController.php
	ContactController.php
	ContentsController.php
	FaqsController.php
	FinishesController.php
	MaterialsController.php
	OrderdetailsController.php
	OrdersController.php
	PagematerialsController.php
	PagesController.php
	ProductsController.php
	ProductsdetailsController.php
	ProductsimagesController.php
	ProductsubsController.php
	ShippingcostsController.php
	ShoppingcartitemsController.php
	SizesController.php
	SliderimagesController.php
	SubcategoriesController.php
	UsersController.php

Discounts

Discounts are manually hardcoded into the source code. They can be found under the orders controller and in the add class. The code reads as so, if the gross total is larger than 50 but less than 100 the discount is 5%, if it's larger than 100 but smaller than 150 it's 15%, if it's larger than 150 but smaller than 300 it's 25% etc all the way up to 40% discount once it's over \$1000



AUSTRALIA'S BIGGEST AND BEST RANGE OF LASER CUT DIY SUPPLIES

```
//check discount
if(($this->request->data['grosstotal'])>50 &($this->request->data['grosstotal'])<100){
    $this->request->data['discount']=0.05;
}
elseif (($this->request->data['grosstotal'])>99 &($this->request->data['grosstotal'])<150) {
    $this->request->data['discount']=0.15;
}
elseif (($this->request->data['grosstotal'])>=150 &($this->request->data['grosstotal'])<=300) {
    $this->request->data['discount']=0.25;
}
elseif (($this->request->data['grosstotal'])<1000 &($this->request->data['grosstotal'])>300) {
    $this->request->data['discount']=0.35;
}
elseif (($this->request->data['grosstotal'])>=1000) {
    $this->request->data['discount']=0.40;
}
//shipping cost
if (($this->request->data['grosstotal'])>150 && $countrytype=='Australia' ){
    $this->request->data['shippingmethod']='Express Post';
}
if (($this->request->data['grosstotal'])>250 && $countrytype=='International' ){
    $this->request->data['shippingmethod']='Express Post';
}
if($countrytype=='International'&&$this->request->session()->read('shipmethod')=='Express Post'){
    $this->request->data['shipping_cost']=0;
    $this->request->data['status']='WaitingForQuota';
}else{
    foreach ($shippingcosts as $shippingcost) {
        if($shippingcost['amount_from']<=$this->request->data['grosstotal'] && $shippingcost['amount_to']>=$this->request->data['grosstotal'] ]
        && $shippingcost['upgrade_type']==($this->request->data['shippingmethod']) && $shippingcost['type']==$countrytype){
            $this->request->data['shipping_cost']= $shippingcost['price'];
        }
    }
}
$this->request->data['status']='unpaid';
}
```



Templates can be located in the template folder, every page has a template associated to it, and furthermore some pages have a specific template for them such as myaccount which has a template stored in src ==> Template ==> Pages ==> myaccount.ctp
















Cartproducts
Categories
Colors
Contact
Contents
Element
Email
Error
Faqs
Finishes
Layout
Materials
Orderdetails
Orders
Pagematerials
Pages
Products
Productsdetails
Productsimages
Productsubs
Shippingcosts
Shoppingcartitems
Sizes
Sliderimages
Subcategories
Users









CSS

Custom CSS can be found in the custom.css file. This is found by going to Webroot==> CSS ==> Custom.css path. From here any CSS class that has been used can be found. There are several major changes from the bootstrap template done in the CSS folder but it's all fairly straight forward.























AUSTRALIA'S BIGGEST AND BEST RANGE OF LASER CUT DIY SUPPLIES

	Name
	bin
	config
	logs
	plugins
	src
	tests
	tmp
	vendor
	webroot
	composer.json
	composer.lock
	index.php
	phpunit.xml.dist
	README.md
	src.zip

	Name
	css
	fonts
	img
	js
	error_log
	favicon.bmp
	favicon.ico
	index.php



AUSTRALIA'S BIGGEST AND BEST RANGE OF LASER CUT DIY SUPPLIES

	Name
	again.css
	base.css
	bootstrap-theme.css
	bootstrap-theme.min.css
	bootstrap.css
	bootstrap.min.css
	cake.css
	coming-soon-social.css
	custom.css
	icomoon-social.css
	leaflet.css
	leaflet.ie.css
	lightslider.css
	main-green.css
	main-grey.css
	main-orange.css
	main-red.css
	main.css
	styles.css
	styles1.css

Business rule for materials and colours

The way crafty cuts laser operates is a little bit unique, because they make to order the system has been specifically designed to allow this. Everything starts from the base material, in Acrylics and Paper the material and colour are one and the same. A product can have one Material but many colours, if one colour is out of stock for any one product it is out of stock for all products.

Suggested improvements for future development

Some suggestions for future development are as follows:

When looking at sizes, shapes, colours etc, they are displayed in the order that their ID are in.

Example: If I add Red it is ID #1 if I then add black it is now ID #2 if I delete Black and add Green it is now ID #3. This doesn't really effect anything when it's colours which don't need to be ordered but sizes are affected.

If the owners would like to add a smaller size it is impossible for her to reorder the sizes without recreating the size list. Therefore it is imperritive for it to be done correctly the first time. This can be improved to allow for ID's to be changed to something like Order # instead.