

A black and white photograph showing two men walking down a set of wide stone steps. The man on the left is wearing a dark, long-sleeved sweater over a collared shirt, blue jeans, and glasses. The man on the right is wearing a light-colored t-shirt with a large 'X' graphic and the letters 'F' and 'B' on it, paired with dark jeans and sneakers. They are both looking towards the right. In the background, other people are visible on the stairs.

Placement report

Stuart Whitehead

Academic year 2013–14

Industrial placement

BSc Computer Science placement undertaken by Stuart Whitehead
from 9th July 2013 to 19th September 2014.

EQ Design

Mark Kemp Creative Director
292 Rosemount Place
Aberdeen AB25 2YA
01224 622277
mark@eqdesign.co.uk
eqdesign.co.uk

Robert Gordon University

Kit-Ying Hui Lecturer
Garthdee Road
Aberdeen AB10 7QB
01224 262707
k.hui@rgu.ac.uk
rgu.ac.uk

Contents

Introduction 1

Project workflow 5

Technologies 9

Projects 19

Summary 45

Other opportunities 51

Thanks 57

References 59

Introduction

Our society has an undeniable desire to share information and arguably, the internet is the epitome of this. Other communication mediums (like the printed word) have a rich heritage stretching many centuries, but compared to that, the World Wide Web is an infant. Sir Tim Berners-Lee only invented it in 1989 and despite its young age, it has connected the whole globe—a feat no other medium has achieved in such a short time frame.

It is this digital era in which my generation has been brought up, and it's proven to be particularly captivating. The new challenges presented by the internet have led me to study Computer Science at Robert Gordon University and to my web development placement at EQ Design.

The main purpose of this report is to fulfil the requirements of my placement, but I'd also like to share my experiences with anyone interested in web development or the creative industries. For this reason, I've been comprehensive in every aspect of my employment.

Europe began using metal printing plates in 1452 (Ambrose and Harris 2007 p. 16) and similar techniques are still in use today. Sir Tim Berners-Lee wrote the original proposal for a 'global hypertext project' in March 1989 before starting development of the World Wide Web program in October 1990 (Connolly 2000).

The road to my placement

I've had a passion for programming since I began experimenting with it in 2003, with design being an additional interest. By the end of secondary school, I had attained the highest grade possible in Computing, Graphic Communication and Physics. I found that the natural progression for myself was to study Computer Science at university.

Before starting my placement, I had a good knowledge of Computer Science principles as taught by Robert Gordon University's School of Computing. These principles included core concepts in object-oriented programming, classic programming algorithms, structured query language and set theory (amongst other computing-based mathematics). By mixing these skills with web-based project work from school and university modules, I had a solid foundation from which to start my placement.

My industrial placement has been undertaken at Aberdeen-based brand and design agency EQ Design. Their staff roster is comparatively small at ten employees, but it's advantageous in the fast-paced design industry because they can be more reactive to change. It was also beneficial for myself because I was able to experience a good cross-section of their business.

The road towards my employment began in the year prior to my placement starting. I responded to an online post by EQ, who were offering work experience in web development. After a casual interview, they invited me to join their web team on a flexible basis. Eight months later—when I was ready to start my placement—I was still very much a part of that team. Agreeably, the next step was to bring me in full-time, and so I was offered a placement opportunity with them.

EQ Design

EQ's roots lie in traditional advertising and brand development, but just like myself, the company was born into the emerging digital era of the 1990s. To maintain a reputation as leaders in their field, EQ has had to embrace the rapidly changing landscape of design. The increasing popularity of the internet has impacted the design sector more so than any other, and for this reason, web development now plays a large role in their offering.

The business itself operates around a flat hierarchy and although the team is small, the combined experience is vast. By using their industry experience, the team can manage, design or develop any aspect of a client's brand. These clients are typically based in the north-east of Scotland, however many are multinational companies giving EQ's work a global reach.

It would be incorrect to label them as a service provider; they do not work for clients, but instead work in partnership with them. EQ Design are brand partners.

This ethos can be seen across every aspect of every project, and it's a great way to work. Just like any other relationship, the partnership between EQ and their clients is dependent on trust and respect, and this is reflected in expectations of the staff.

They're empowered to make their own decisions and are rewarded for good work. For example, there's no set dress code and the atmosphere is very relaxed. There's also a social committee, who've got a generous budget to organise social events and days out. The team have been rewarded with this freedom because they're trusted to produce great work, and I think that's liberating.

Netcraft (2014) has found that the number of hostnames on the internet has increased from only 19,732 in August 1995 to 975,262,468 in May 2014. That meteoric increase shows just how quickly the internet has gripped ahold of the world.

Role and Responsibilities

Just like the other staff members, I was trusted to make objective decisions and to produce great work throughout my placement. Websites are becoming technologically more diverse, so I've found my background in computer science to be useful in this role.

I worked with two other developers and between us, we had a wide variety of skills. While I didn't have an explicit role (such as front-end, back-end or system administration) my strengths lie in programme logic and back-end code, so I found myself working mainly on this aspect of projects. My responsibilities as a web developer can be split into two categories.

Technical

As a developer, my overall role was to aid in the creation of websites. This covered a variety of different tasks, from front-end tweaks to the complete configuration of website back-ends.

It was my responsibility to work on tasks where and when I was required, which included setting up development environments; configuring content management systems and developing bespoke application code.

Administrative

As with any other business, there were administrative processes to follow. I was expected to record time spent on jobs with time sheets, which the directors can then use to forecast for future projects.

It was important that I communicated clearly and effectively between colleagues, mainly through email or project management software. I was also required to liaise with clients in a professional and courteous manner.

Project workflow

The experience gained from previous projects has allowed EQ to develop a solid project workflow. They've found that a partnership between themselves and their clients produce the best and most effective work, so each project focuses on embracing this relationship.

It starts with a brief

The majority of work undertaken by EQ is initiated by their clients. This work will come in the form of a brief, outlining the requirements of an individual project. EQ will take this information and create a proposal which will discuss how they would handle the project. It will also detail some relevant pieces of work and it will provide an indicative guide to the scope and cost, based on previous experience.

Discovery

Once EQ has been handed a project, the first step is a 'Discovery Phase,' and it's really important. It's impossible to develop an effective solution if they know nothing about the client or their objectives. In essence, this all comes down to talking. Whether that's in a meeting or over a coffee, it's important to gain an honest perspective of the business from the people that work in it. Remaining focused at this stage will help to direct the project later on.

Content strategy

The discovery phase will generate a lot of data—both about the business itself, and the services or products which they offer. Each of the subsequent stages of design and development are informed by this content, so it's now important to make sense of the information.

To enable this stage to be effective, it's important to liaise internally between colleagues and externally between the client. The content will be refined multiple times until two documents can be produced. The first, a structure document, describes the overall hierarchy of pages. The second, a content document, contains all of the content for those pages. A combination of these two will allow anyone involved in the project to understand the structure of content at a glance, and it will highlight if and where any content is lacking.

EQ places great importance on getting to know their clients. A good example of this was a discovery event which EQ hosted for a new client, Read Cased Hole. A relaxed mixology lesson allowed them to gain an honest perspective through the eyes of the most important people, the employees.

Designing for the web

Web designers traditionally created Photoshop mock-ups for clients, which show the content in place and how the whole page will look. However, this technique doesn't work well with modern-day web development as a result of responsive design. Different devices have different design constraints, so a more flexible approach needs to be taken. EQ have two creative devices which make this process easier.

First of all, the content documents are used as a basis for wireframes. Wireframes do not show the layout of a page, but instead represent the hierarchy of content. The actual content can move around as the screen size changes, but the order of importance remains the same.

The second device is a style guide. Style guides are produced by designers and describe how individual components of a webpage should be styled up. Front-end developers will then take these guides and apply the styles to HTML mark-up. Most importantly though, the developers are free to work within the constraints of different devices and modify the styles appropriately. It's common to see designers and front-end developers collaborating at this stage.

Iterative development

EQ maintains an iterative approach to their web projects, meaning a website is never finished. After the initial build is complete, it's important to monitor and review a website's performance. A website should continue to fulfil the client's objectives but if monitoring shows that the website is performing below expectations or goals, iterative improvements can be made. These changes will then be monitored, and the cycle will continue.

From a technical perspective, software requires continual updating. As part of this iterative approach, software will be inspected at regular intervals and updated if required.



Technologies

EQ Design develops code exclusively for the Internet. Their websites are powered by community-driven software and in particular, a traditional LAMP stack is used.

Readers of this report may already have a knowledge of such technologies, but I'd still like to discuss them. This is to provide insight on my role and what technologies a design agency may use.

Languages

I used both front and back-end web languages during my placement. Each language has its own role to play, but as Jeremy Keith (2010) suggests, they are all tied back to the Hyper Text Markup Language (HTML) of a webpage.

HTML5

HTML is a simplistic language which at an abstract level uses pairs of tags to describe content. Under the HTML5 specification written by the World Wide Web Consortium, there are specific tags and attributes to describe most content scenarios (Keith 2010). Writing so-called 'semantic code' not only allows search engines to crawl content effectively, but it also allows future developers to understand the work.

Web browsers are forgiving when it comes to rendering HTML, but that's not to say developers can be sloppy. There are accepted best practices for marking-up specific types of content, especially when it comes to writing machine-readable meta data. Microformats use existing HTML tags and attributes to add meaningful information to content, which can then be parsed by search engines (Lawson 2011).

CSS3

CSS is used to apply styles to a web page and can be included in any of three ways: an external stylesheet, an internal stylesheet or an inline attribute. CSS selectors are used to reference HTML components of a web page, of which dimensions, layout, typography and colours can then be modified.

As with HTML, it's important to write meaningful CSS code, which is usually achieved through descriptive selectors. EQ use a modified version of the BEM (Block, Element, Modifier) syntax to reference elements. This allows a developer to understand what a selector is

In CSS, the specificity of a selector is how targeted it is, and it can cause issues with styles overwriting others. The BEM syntax helps to alleviate this, as most selectors will have the same specificity. It also makes code easier to read
(Roberts 2013).

targeting; and prevents other styles from being overwritten. This language looks basic on the surface, but it is in fact very powerful, and plays a large role in Responsive Web Design. Modern CSS rendering engines can draw complex graphics on the fly, such as box shadows and 3D shapes, as well as powering smooth animations.

CSS is rendered by web browser rendering engines. How these engines render styles is down to their individual vendors, however they must follow the Level 3 CSS specification (again written by the World Wide Web Consortium) to ensure compatibility. Some CSS features are considered to be experimental so vendor prefixes are required until they are firmed out.

Sass

Sass (humourously) stands for 'Syntactically awesome stylesheets' and blurs the line between language and tool. CSS is a static language, so as a CSS preprocessor, Sass provides developers with some dynamic programming structures.

There are two different syntaxes—sass and scss. Scss is more like plain CSS as it uses brackets, whereas sass is dependent on indentation, which I think is more succinct and easier to understand. When run, the Sass compiler will render code down to CSS, and can also handle different output formats: nested, expanded, compact and compressed (which is great for production code.)

The features of Sass are very useful in the development of CSS. Sass's variables support six main data types (number, string, color, boolean, null, lists) and their values will be evaluated at runtime. Expressions (if, for, each, while) can be used to dynamically generate code and mixins are styles which can be re-used throughout a file. It's even possible to define reusable functions.

PHP

PHP is a general-purpose server-side programming language. The abbreviation stands for Hypertext Preprocessor which harks back to its origin as a tool to dynamically generate HTML. While PHP is still used to create HTML, its role in modern-day web development is much more broad. Its architecture allows for great flexibility, and can run anything from simple procedural scripts to large object-oriented applications and powerful frameworks (PHP.net 2014).

When it comes to PHP's data types, it is generally a weakly-typed language. While it does support a good range of data types (boolean, integer, floating-point, string, array, object, null), PHP doesn't support explicit type declaration. That makes it legal to assign a string value to a variable containing an integer, for example. But in saying that, PHP supports what it calls 'type hinting.' Type hinting can force a parameter to be of a specific class.

Quirks like these have given PHP somewhat of a bad name amongst developers. Compared to languages like Java or Python, the PHP core isn't consistent. But it is possible to develop robust, clean code with it. It is a hugely popular language thanks to the pragmatic functions it provides, powering over 200 million websites (Netcraft 2013). These functions make common tasks like form handling, data persistence and template rendering straightforward.

A good example of a robust PHP application is the content management system which EQ uses. Craft CMS is powered by PHP, and is built on top of the Yii framework (which has been developed strictly with object-oriented principles). The majority of PHP development I carried out during my placement was for Craft, specifically custom plugins for it.

Javascript

Javascript is the most common scripting language on the web and has been around since its inception by Brendan Eich in 1995. In 1996, Eich took Javascript to the ECMA body which led to the ECMAScript standard, and has since been used as the basis for implementations of Javascript (Flanagan 2011).

The primary use of Javascript in web development is to dynamically modify the HTML and CSS of a web page. It is tightly bound to HTML and subsequently to CSS through the Document Object Model, which represents all nodes of an HTML tree. It is a weakly-typed functional programming language and has got three very recognisable characteristics:

1. Being a weakly-typed language, type declarations are not required. For example, this makes it legal to assign an integer value to what was a string.
2. Javascript's event-driven architecture is ideal for asynchronous applications. Callback handlers call code after an event, so it's common to see functions nested deep within each other.
3. This language implements prototypal object inheritance, making it possible to embrace some object-orientated programming principles. Every Javascript Object (except the base Object) has a prototype.

While other browsers implement Javascript, Microsoft's browsers (up to Internet Explorer 9) implement JScript, Microsoft's own dialect of ECMAScript. As a result, there are some inconsistencies between Javascript engines—the jQuery library is used at EQ to provide a uniform cross-browser solution.

The number of Github repositories written in Javascript trumps any others at around 350,000, compared to Ruby in second place with 260,000, showing just how popular the language is. Javascript data was generated using Adam Bard's method (<http://adambard.com/blog/top-github-languages-for-2013-so-far/>) with Google BigQuery.

Tools

To make the most of those languages, I used a variety of different applications and frameworks. These are the specific tools which I utilised during my placement:

Git

Git is one of the most important tools which the development team at EQ uses. It is a complete command-line version control system, and is used to track changes to files. As an individual, it is important to track local changes to code. By using Git, a developer can test variations and revert to a previous state if these were unsuccessful. Ultimately, it ensures that work isn't lost.

Git is doubly important for use in a team and it integrates well with other services. One such service is Beanstalk, which EQ use as a central collaboration point and to deploy code live.

Craft CMS

EQ develops websites around their client's content, and to manage this content effectively, a content management system is used. One such PHP application is Craft, and it's hugely flexible. It can run on EQ's technology stack and allows for pretty much all of their use-cases, making this their content management system of choice.

Craft can store virtually any type of content in a structure of the developer's choosing. It also has a powerful 'relationship engine,' which allows for complex content relationships. This is all controlled through a simple control panel, making it easy for clients to use.

It's highly stable thanks to its foundation on the Yii PHP framework, and templating is simple with the Twig templating engine. If a use-case is still not covered, Craft's plugin architecture allows developers write custom application code for any aspect of the system.

Node.js

Although Craft CMS is hugely flexible, it may not always be the appropriate tool for a project. Throughout my placement, I've used node.js for smaller-scale, specific tasks. Since its public release in 2009, node.js has proven itself to be great for network-based applications. It allows developers tight control over the lower-level abstractions of the internet.

Given that it's built on Google's V8 javascript engine, node.js is extremely fast and is perfect for handling asynchronous communication. Its event-driven architecture also makes it an ideal tool for real-time web applications using Web Sockets.

Gulp.js

Node.js was built to be highly portable, making it just as easy to run on a local machine as it is a server. Not only has this portability given developers freedom to run local development environments, but it's allowed them to 'hack' together interesting tools.

One such tool is Gulp.js, which is a task runner used for the optimisation of website assets. There are many different tasks which can be run, but a typical goal is to combine common files into one, to reduce HTTP requests and overall load times. For example, a folder of five CSS files could be combined into one file, then minified (reduced to the bare essentials) to reduce its overall file size.

Sublime Text 2

Sublime Text 2 is my text editor of choice. Its interface is really simple, it has powerful shortcuts and functions and is extensible through the use of a Python-based API. I used it for all code development during my placement. If the code became any more complex (such as a complete object-oriented PHP application), an integrated development environment such as PHP Storm would be more appropriate.



Services

EQ Design makes use of many services for the end-to-end completion of a project. These services are used across the whole development cycle, and range from scheduling to code deployment.

Basecamp Classic

Effective administration makes a big difference when working on a project as it ensures time and costs are being adhered to. EQ use the service Basecamp to manage all administration—from recording communication with clients to storing briefs, proposals and content documents. As this service is internet-based, the team at EQ can access it from work or at home, meaning they can continue to work no matter where they are.

Beanstalk

Git is great at tracking code changes, however it's difficult to enable collaboration and to deploy code using it alone. To publish code, we use the service Beanstalk, which integrates tightly with Git.

We use Beanstalk for two reasons. Firstly, it acts as a central repository. This keeps code under control and prevents anything from being accidentally overwritten. Secondly, it handles deployments over SSH File Transfer Protocol (SFTP). When a code edit is 'pushed' by a developer, Beanstalk deploys only the changed files. It also provides handy features for comparing Git commits, so the team tell what code is being deployed at a glance.

Linode

To push any website live, there must be a web server to host it. These servers can come in many varieties, but as previously mentioned, EQ favours a traditional LAMP (Linux, Apache, mySQL and PHP) stack. These are the four basic technologies required to serve EQ's flavour of Web projects.

EQ uses Linode's virtual private servers to host their websites. Their business is simple: they maintain the server infrastructure, while EQ can install the Linux operating system of their choosing and any software they may need (such as the LAMP stack required).

They supply a great service as it's possible to spawn virtual private servers in many different locations across the world, reducing overall latencies. They also have a good reputation for their support, so if there's a fault, they will be on hand.

DNSimple

The internet-wide domain name system (DNS) uses nameservers to resolve human-readable domain names (like google.co.uk) into IP addresses (like <http://74.125.224.72>) which point to web servers.

When spinning up a Linode server, we're assigned one of these IP addresses. To convert that into a URL, we use the DNS service, DNSimple. This gives us total control over a domain name's DNS settings, allowing us to control domain and sub-domain names, email records and URL redirects, all from a simple web-based control panel.

Heroku

While it is possible to serve node.js applications from virtual private servers like Linode's, it is much easier and faster to make use of a Platform as a service (PaaS). A PaaS is a complete technology stack which will run an application and it's perfectly suited to node.js.

Throughout my placement, I made use of Heroku for my small scale projects. The main reason for this is that a development 'dyno' is free. The Heroku system supports web sockets for real-time web communication, and the platform will take care of installing missing dependencies too.



A black and white photograph showing three people working together at a table. In the center, a young man with short hair and a beard, wearing a plaid shirt over a white t-shirt with the word 'REAL' printed on it, is smiling broadly. To his right, a woman with glasses and dark hair is looking down at something on the table. On the far left, another person's head and shoulders are visible, also looking down. They appear to be working on a laptop or a small screen. The background is plain and light.

Projects

From the moment I started working with the team, I felt that both my technical skills and personal opinions were respected. This mutual respect and trust worked really in a placement environment, and I was exposed to a good cross-section of projects.

Thorntons Property

The first commercial project which I was a part of was the complete redevelopment of a property website for Thorntons Property.

Thorntons are a prominent solicitor-based estate agency in Tayside, where they've got four branches—Dundee, Perth, Arbroath and Forfar. They have been considered a market leader in the area for the last 20 years.

The clients selling and buying property with Thorntons are placed mid to high end of the market, exchanging properties valued at £150,000 or above. Thorntons recognise that a high percentage of their business is driven from the website, thorntons-property.co.uk, and so EQ Design were enlisted to improve its effectiveness.

The brief

In line with EQ's project workflow, Thorntons submitted a brief prior to a proposal being drawn up. They stipulated three main objectives of a new website—the following is an excerpt from their brief.

"The TPS website needs to serve several purposes. In order of priority, these are:

- To provide a clean and clear property search facility in order that visitors—buyers and sellers—can quickly and easily access our property database to find property for sale (inter-action required with Vebra Live which will drive the search function.)
- To promote and sell the services of TPS to potential sellers and buyers; engage their interest and encourage them to use our services through click throughs to view; request more details and request a valuation/quote.
- To deftly cross-sell other property related to Thorntons services—e.g. mortgages and legal conveyancing services—once the visitor has found a property."

One important point made in the brief was that a third-party service, Vebra Live, would be used to drive the search function. This particular aspect suited my personal skills, which includes the programming knowledge which I have been taught at Robert Gordon University.



Live search plugin

The client's previous website used HTML forms to submit search criteria. This traditional method is stable and is used on websites across the world, however, website users now expect a more feedback-rich experience.

In response to this, EQ proposed a live search, which makes use of the now widely adopted AJAX standard. AJAX stands for 'Asynchronous Javascript And XML' and allows a web page to load new data without reloading the whole document—ideal for retrieving search results. During the early stages of development I was tasked with building a prototype to help explain this concept to Thorntons.

The prototype used Javascript to make an AJAX call to a development server, which generated a response based on a sub-set of real data. These results would be displayed automatically, giving the impression of a real-time interface. To save work further down the line, the prototype used the same server-side software which the complete project would. This prototype was then shown to the client as an example of the technology, and led to the proposal being agreed upon.

Once the proposal had been signed by both parties, I advanced the prototype into a production-ready plugin. As this was to be used on a production site, the main objectives of the code were robustness and efficiency. By condensing this into a jQuery plugin, my code was able to use jQuery's solid cross-browser methods.

One main challenge for the production code was enabling the script to track these input states (this was especially important for the pagination aspect.) To keep track of search criteria and result offsets, I made use of client-side cookies.

Vebra import script

Another challenging aspect of this project was the use of a third party API. The API was slightly dated and responded with XML, which is a bloated technology and ultimately, generates large file sizes (in comparison to the leaner JSON standard).

The main objective of this script was to poll the API for properties whose data had changed and then import them into the content management system.

This script was written in PHP because it provides pragmatic functions for making HTTP requests and handling XML nodes. To ensure that this script was run at regular intervals, a Unix CRON job was set up on the production server to trigger the request.



I developed the nickname 'El Vebra' after wrangling with this aspect of the project. As a thanks, the EQ team bought me an 'El Vebra' mask. It's a lighthearted gesture but it showed that my work was appreciated.

Summary

Given that this was my first live project with EQ, I feel that I conducted myself professionally. The requirements were challenging, but by working closely with my colleagues, we created an effective website which adhered to the brief.

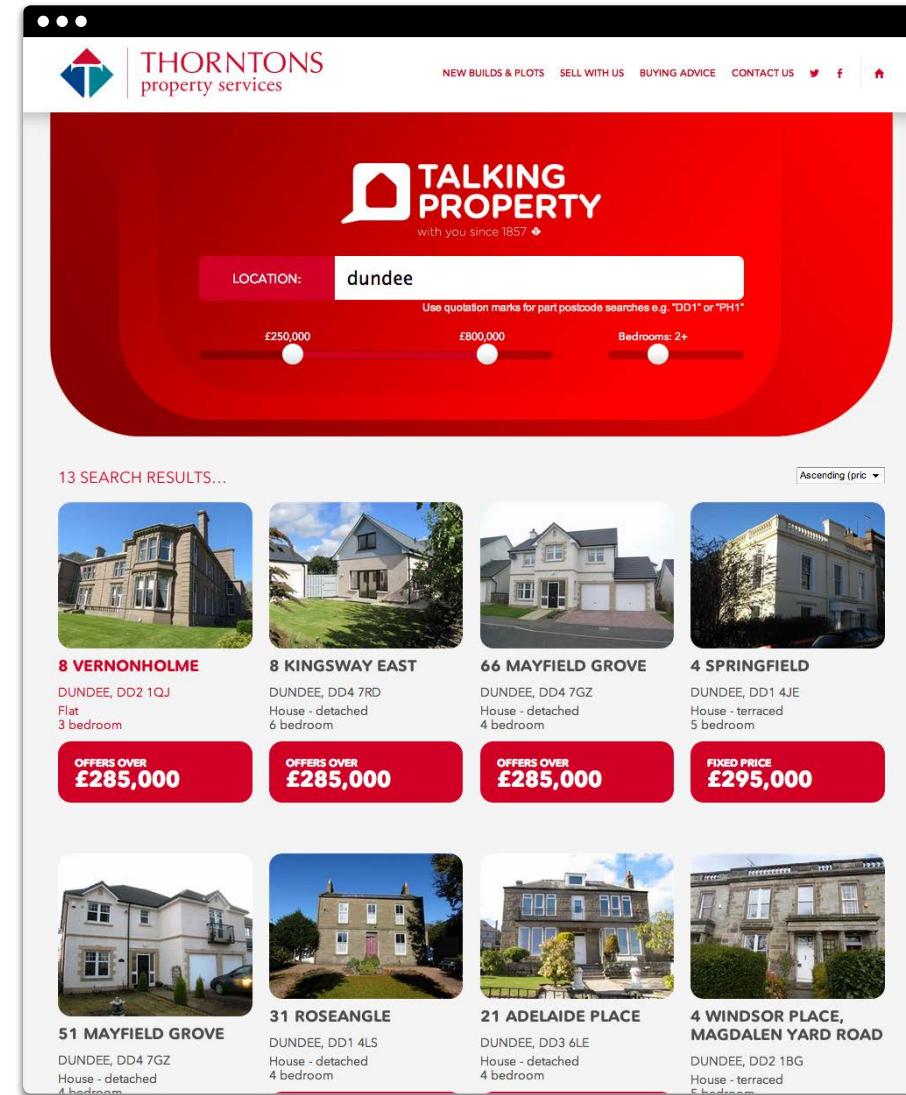
At points throughout the project, I liaised with the client to explain how the system was being developed, and to work on any comments which they had raised. I think that this coupled with my production-ready code earned the respect of my colleagues and set the tone for the remainder of my placement.

"There is no substitute for a real world problem which both interests and challenges a developer's mind. When Stuart started at EQ, we had just the thing for him to get his teeth into. We had a requirement to integrate XML data from a third party API into an ExpressionEngine based property website.

No existing library met our use case which meant we needed a custom built PHP solution to get the data from the API into our system. We had to query this data 11 times throughout a day's typical working hours as well as a full data dump in the early hours of each day.

Stuart delivered a solution which has been running without issue for over 12 months now. His natural ability with code and skills gained on his course are clear to see."

—Mark McAulay Developer



Simpson Forsyth

The second website I'd like to share is for Simpson Forsyth, who are an Aberdeen-based chartered accountancy firm. They offer typical accountancy and taxation services as well as payroll and corporate finance. Using their experience and knowledge, Simpson Forsyth have the ability to advise on any aspect of a business—from mergers and acquisitions to valuations and business disposals.

They are an existing client of EQ, with their previous website being a past EQ project. Since then, web technologies had greatly improved, making their old site outdated. Using various analytics tools and services, EQ created a review of the website, noting what worked well and what could be improved. Using this evidence, they approached Simpson Forsyth and a redevelopment was agreed upon.

The brief

This project was initiated by EQ, meaning that there wasn't a brief as such. Instead, the recommendations noted in the review formed the basis of a proposal.

Starting off, it was noted that the copy across the site wasn't effective, both for the user and search engines. EQ understand that users respond better to a narrative—rather than segregated silos of content—and so to improve on this, a copywriter was to be involved. A copywriter would help to normalise such content and create a consistent tone of voice across the whole site.

To further improve on the content, EQ suggested re-working page hierarchies. This would create a user journey through their content, compared to the brochure-like experience which the old website offered. One limiting factor with this, however, was that the old site used WordPress. While great for blogs, it's not ideal software for more complicated content structures and for that reason, EQ suggested using a different, more user-friendly content management system.

Allowing for a more complex site structure wasn't the only benefit to a new content management system. It would also allow for better handling of URL formats, page titles and page descriptions, which was another aspect which the old site was found to be lacking. These are important for search engine rankings.

As with all of EQ's new projects, Simpson Forsyth's website redevelopment would be responsive. Careful consideration would be taken at every stage for a cross-device user experience.



Configuring server environments

During the development of Simpson Forsyth, I was tasked with configuring the hosting environments for the first time. To do this, I was given administration privileges for most of EQ's hosting stack, and that carries with it many responsibilities. Configuring the stack followed this flow:

Setting up the server

This is one of the most important steps in hosting a website because without a server, there's nothing to host it on. EQ maintain two virtual private servers—a production server and a staging server. While permissions and configurations are slightly different between them, the process is the same.

They use the open-source control panel Webmin. Preparing the server for a new site is as simple as creating a new virtual host with Webmin, which will take care of everything else. The only thing left to do is to point a domain name at the server with DNSimple.

Initialising a Git repository

Now that hosting was enabled on the server, I had to allow files to be pushed to it. EQ use a workflow built around Git using the service Beanstalk. The first step is to create a new repository on Beanstalk, which is done through a web-based interface. Then I initialised a local Git repository and added the Beanstalk repository as a remote. Now that they're hooked up, I could push files locally to Beanstalk.

Once the files were under Beanstalk's control, I could set up a deployment to the server. Again using the web-based interface, I set up a deployment using SSH File Transfer Protocol (SFTP). This means that anytime a file is pushed locally to Beanstalk, it will then be automatically deployed to the server.

Configuring Craft

Along with setting up the hosting environments, I was tasked with configuring Craft for this project. I had been experimenting with Craft over the previous month—it was brand new and EQ needed to test how useful it was. It was in fact powerful and easy-to-use, so the decision was made to use it on a live project for the first time. I was chosen to configure Craft because I had spent the most time getting to know it.

Since the server and Git workflow had been set up, configuring Craft was a fairly straight forward affair. There were two main points to consider: the system files and the database. First of all, I installed Craft locally, which let me configure the multi-environment settings and other general settings. Once everything was up and running, I committed the Craft files to Git and push them to Beanstalk. To finally get Craft running on the remote servers, all I had to do was import the database as everything else had been deployed automatically.

To kickstart the build, I created the necessary fields within Craft and pre-populated these with content from the content documents. This then allowed EQ's front-end developer to build out the templates.

Client training

Once the build of the site had been finished, the staff at Simpson Forsyth had to be trained on using Craft. Craft itself is easy to use, but being shown around the control panel is always welcome. Since I had configured most of the back-end, I was given the responsibility of training the staff.

I travelled by myself to the client's office where I gave a brief training session. This was the first time I had one-to-one time with a client.

Summary

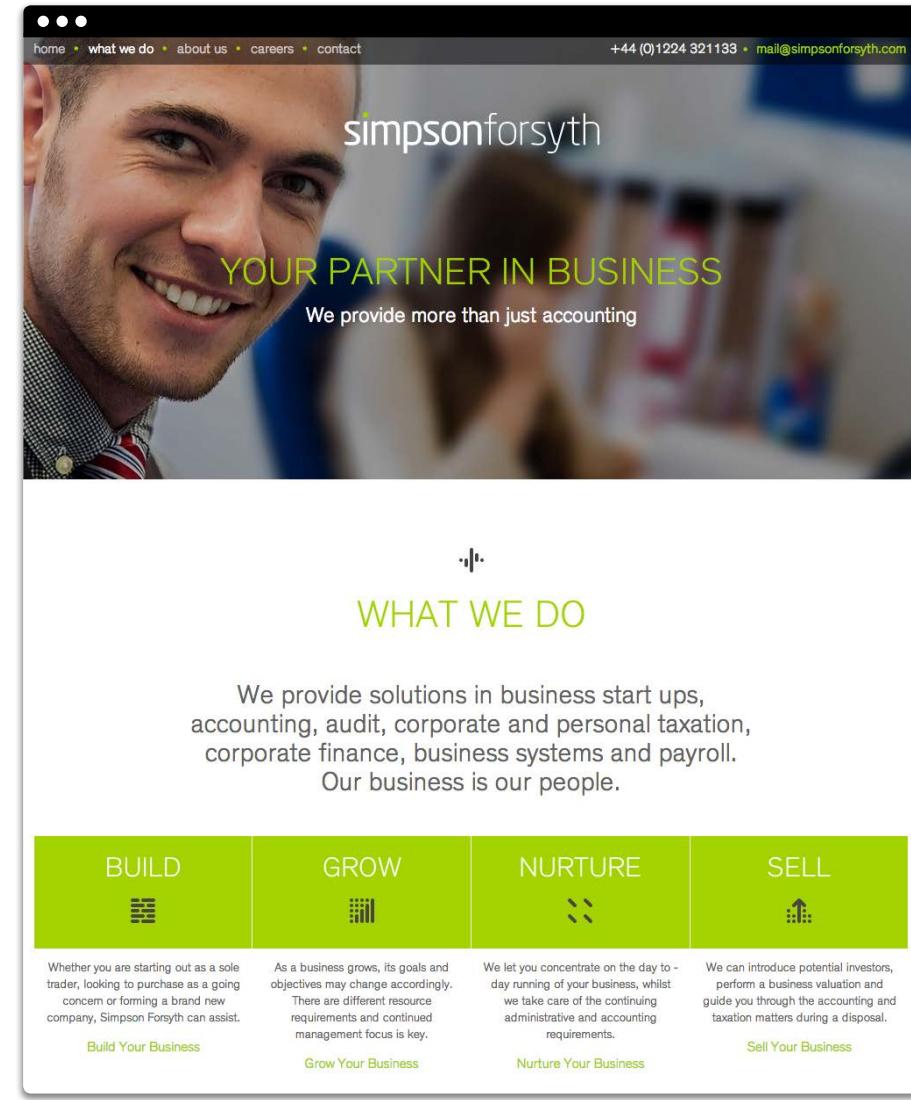
Like I touched on, I was allowed a considerable degree of freedom on this project. Of course, I was supervised along the way, but I was generally left to my own devices to complete tasks. Reflecting on what was achieved, I certainly improved on many skills.

In particular, I learnt a lot about the main features of Craft. Given this new knowledge, I was able to bring experience with me into future Craft projects, streamlining the process. I also developed a confidence when dealing with clients, a result of being trusted to travel to a client's office and train them by myself.

"All web projects at EQ include an hour of one-on-one CMS training with either myself, or one of our developers. With the Simpson Forsyth project, it was the natural and correct choice to have Stuart give this training given that he worked independently on the back-end build of the site using 'Craft'—which was new to us!"

Initially, I intended to attend the training also so that I received training in Craft at the same time as the client, however due to commitments with another client, Stuart went to Simpson Forsyth himself. Our client was pleased with the training and has since been in touch with Stuart when she has had any queries regarding the CMS (although these are few and far between). Stuart is incredibly personable and has shown in his client-side experience with Simpson Forsyth that he can talk to a client in a way they will understand based on their respective technical knowledge."

—Kirsten Young Project Manager



The screenshot shows the Simpson Forsyth website. At the top, there is a navigation bar with links for 'home', 'what we do', 'about us', 'careers', and 'contact'. To the right of the navigation, there are phone and email contact details: +44 (0)1224 321133 and mail@simpsonforsyth.com. The main header features the company name 'simpsonforsyth' in a bold, sans-serif font. Below the header, a large, close-up photograph of a smiling man with short brown hair and a beard, wearing a white shirt and a red striped tie. Overlaid on the photo is the text 'YOUR PARTNER IN BUSINESS' in a large, bold, green font, followed by the subtitle 'We provide more than just accounting' in a smaller, black font. The page then transitions into a section titled 'WHAT WE DO' with four main service categories: 'BUILD', 'GROW', 'NURTURE', and 'SELL'. Each category has a corresponding icon and a brief description below it.

BUILD	GROW	NURTURE	SELL
			
Whether you are starting out as a sole trader, looking to purchase as a going concern or forming a brand new company, Simpson Forsyth can assist. Build Your Business	As a business grows, its goals and objectives may change accordingly. There are different resource requirements and continued management focus is key. Grow Your Business	We let you concentrate on the day-to-day running of your business, whilst we take care of the continuing administrative and accounting requirements. Nurture Your Business	We can introduce potential investors, perform a business valuation and guide you through the accounting and taxation matters during a disposal. Sell Your Business

Douglas-Westwood

A third project I worked on was the redevelopment of a website for Douglas-Westwood. They are a leading provider of market research and consultancy services to the global energy industry, making them an important company in Aberdeen. Their market research generates proprietary data, which they use to develop important insights on many corporate, financial and governmental sectors.

The brief

Before EQ Design became involved, Douglas-Westwood were partnered with a design agency near their head office in Great Yarmouth. While this agency had almost completed a redevelopment of the website, Douglas-Westwood weren't happy with the direction it was heading in. As a result, EQ were briefed on why this site fell short of their expectations. These are some of the key points made:

- The website was too difficult to navigate.
- The information was organised into several layers, but they felt this could be done in a more intelligent way.
- The site had the feel of distinct templates, rather than a continuous user experience.
- The site was too linear.

EQ were asked to redevelop the site in a more clever way. The objective was to convey Douglas-Westwood's breadth of experience, and to do that, we were to portray how their services were related to the sectors in which they research.

My role

By the time development began with this project, I was comfortable with Craft CMS and ready to put it through its paces. Given that Douglas-Westwood wanted their content to be used in an intelligent way, Craft's relationship engine would be the idea tool for the job.

My role in this project was to configure and maintain the content management system; to generate templates based on the content and to devise the relationships using Craft's relations. This is how that role broke down:



Planning

The first stage—before any software was installed—was to work out how the content would be structured in Craft. To do this, I took to EQ's writeboard. Using a pen and static notes, I wrote out each block of content and grouped them based what it was representing (such as a service or client market).

Having a visual representation of the content was helpful, as it allowed me to thoroughly understand what I was working with. Using what I had learnt here, I was able to define relationships between types of content which could then be transferred into Craft.



Content relations in Craft

As with Simpson Forsyth, I created new server environments; I set up Git with Beanstalk and I got Craft installed. Configuring all of these was only the beginning of my involvement in this project. Taking what I had worked on with the writeboard, I had to enable Craft to create meaningful relationships between content silos.

The relationship engine in Craft is one key feature which makes it unique. The developers have created (what they call) element types. These element types represent all sorts of dynamic data (namely assets, users, entries, tags and categories). In object-oriented terms, these element types inherit properties and methods from a base class. They are also all referenced by one table in the database, and this allows them to be related to each other.

In order to allow the front-end developer to hit the ground running, I created the basic template structure of the site. These templates output all of the content stored in Craft (including relations), into the correct hierarchy on a page. The relationships between content are defined in the control panel, but it's in the templates where they become powerful. The following code would produce a list of news articles which are related to the current page being viewed:

```
{% set posts = craft.entries.section('news').  
relatedTo({targetElement: entry}).find() %}  
  
<ul>  
  {% for post in posts %}  
    <li><a href="{{ post.url }}>{{ post.title }}</a></li>  
  {% endfor %}  
</ul>
```

Summary

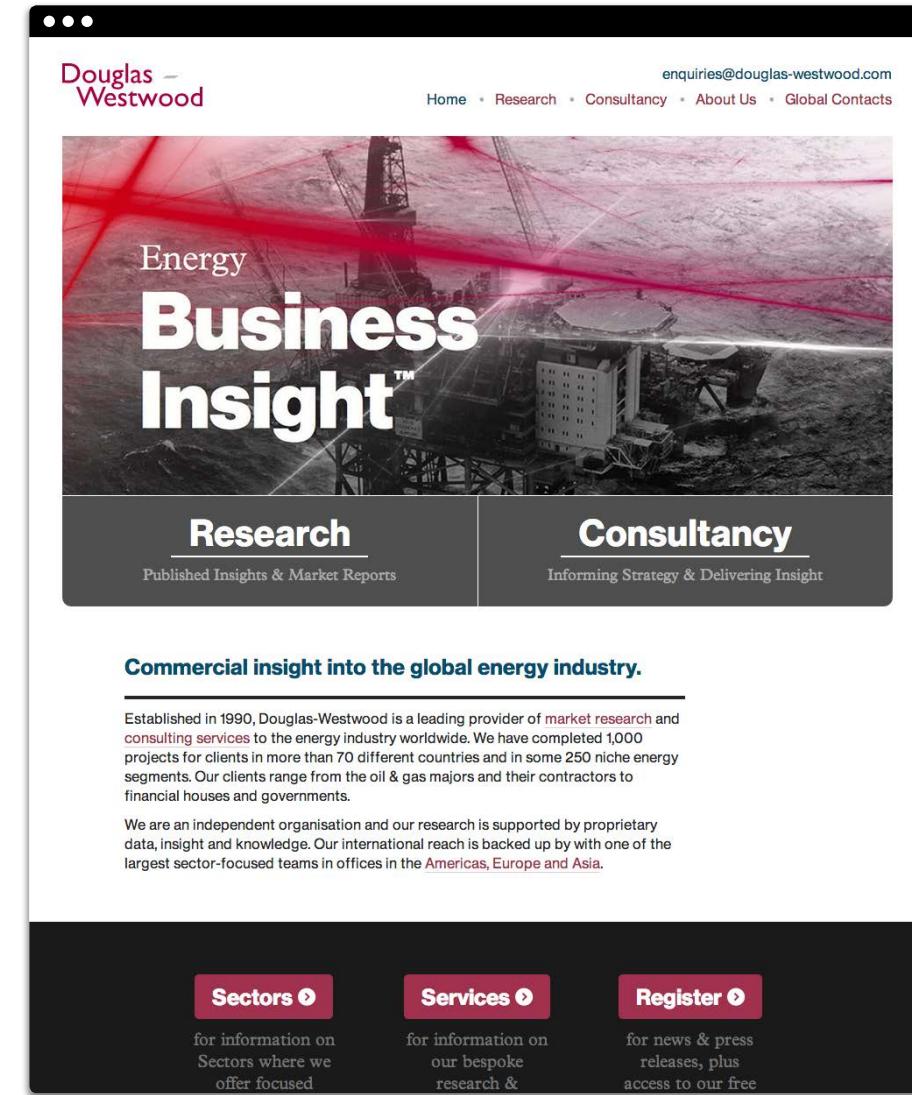
EQ considers this website to be one of its flagship projects, and quite rightly so. It took the whole team's dedication over a prolonged period and like them, I'm proud of the result. Although the sheer quantity of content posed a challenge, I believe that the whole project was handled competently.

Reflecting on my own involvement, I found this project really interesting. I learnt a lot about Craft and it's relationship engine. These are all skills which I can transfer to future projects.

"Stuart's involvement in Douglas Westwood's site structure was absolutely integral to the delivery of the project. Working closely with our developers he knew their business offering inside and out. Only with this level of insight is it possible to work on web projects which truly live up to our clients expectations, and their customers requirements.

Even by our standards this project was incredibly complex. Stuart seemed to understand the relationship between client markets, services and sectors Douglas Westwood work within intuitively—which I must admit is slightly terrifying. This put him in a prime position to build out the CMS and allowed him to succinctly communicate with the client throughout every stage of the project."

—Stewart Ainslie Designer



The screenshot shows the homepage of the Douglas-Westwood Business Insight website. At the top left is the logo "Douglas - Westwood". At the top right is the email address "enquiries@douglas-westwood.com" and a navigation menu with links to "Home", "Research", "Consultancy", "About Us", and "Global Contacts". The main header features the words "Energy", "Business", and "Insight™" in large, bold, white font against a dark background with red light streaks. Below the header is a large image of an industrial facility, likely an oil rig or refinery. The page is divided into two main sections: "Research" (with subtext "Published Insights & Market Reports") and "Consultancy" (with subtext "Informing Strategy & Delivering Insight"). A section titled "Commercial insight into the global energy industry." follows, with text about the company's history and services. At the bottom are three call-to-action buttons: "Sectors" (for information on focused sectors), "Services" (for bespoke research), and "Register" (for news and press releases).

Other work

These three websites are a great summary of my role at EQ, but I did complete many other projects during my time there. Some of these were client sites, and some of them were personal projects. Either way, these are some of the other projects I worked on.

Craft sites

Along with Simpson Forsyth and Douglas-Westwood, I worked on numerous other Craft-driven websites. My role across these was similar to what I've already mentioned, so they don't require extensive explanations.

Caledonian Optical

Caledonian Optical are an interesting client. They operate an optics laboratory in Aberdeen specialising in lenses for glasses. In order to improve exposure for new ranges of lenses, EQ developed a new brand for them and to go along with that, they required a new website and printed collateral. The website was a simple brochureware site with a few interesting content structures, but it was nothing that phased Craft.

John Lawrie

The John Lawrie Group specialise in scrap metal reprocessing and steel trading. They had decided to split the group into two distinct divisions—tubulars and metals. EQ were contracted to rebrand these divisions, and to create print collateral and a new website to support the roll-out. The website had to reflect that there were now two different divisions, so categories were used in Craft to distinguish content between them.

ActivityMix

The website for ActivityMix was split into two phases. The first phase was a small marketing website and needed to be in place for the launch of the brand. The timescale for this phase was short so we had to focus our efforts. The second phase expanded on the website, and included a blog and case studies. Craft's flexibility was clear to see when returning to development for phase two, as we could continue where we left off.



Teaboy app

I was given a lot of freedom when it came to learning new skills. Everyone at EQ appreciates that technology is always improving and without constantly trying new tools and techniques, it's easy to be left behind. One such technology I experimented with is node.js and in particular, using it to harness real-time communication.

There was a running joke naming me 'The Teaboy' throughout my placement. Combining both the joke and my desire to experiment with new technologies, I created the Teaboy app—an online tea and coffee ordering... thing. Although the concept was humorous, the underlying technology was really interesting. The emerging web sockets standard allows a real-time connection to be made between a client browser and a server, and enables either end to send or receive data.

The application itself was really simple—everyone in EQ had their own URL to use. On their own page, they selected tea or coffee; milk or no milk and the amount of sugar they wanted. On the Teaboy's control panel page, the orders would appear in real time. Experimenting like this has allowed me to learn about cutting-edge technologies, which one day may become widely used.



Web server configuration

When working on a web project, there's many aspects that need to be considered, such as the creation and structuring of content; development of front-end and back-end code and configuring content management software. No matter the project, they will all need to be hosted on a web server and configuring one is a daunting thought, at least at first.

To build experience with such tasks, I began experimenting locally on my Macbook Pro. The Mac OS X operating system is developed on top of Unix, meaning it has similar characteristics to the Linux distributions (also developed on top of Unix) which are commonly found on web servers. In particular, I was able to learn and practice bash commands with Mac OS X's Terminal application.

Installing and configuring a web server from scratch is a totally different challenge. To allow experimentation with software like this, EQ have a Raspberry Pi, which is a small and inexpensive Linux-based computer. Using that, I could install different Linux distributions and web server software (such as Apache or Nginx) without the danger of breaking or corrupting expensive hardware.

```
pi@webbox ~ $ sudo apt-get update
pi@webbox ~ $ sudo apt-get upgrade
pi@webbox ~ $ sudo apt-get install nginx
pi@webbox ~ $ sudo apt-get install php5-cli php5-cgi php5-fpm
pi@webbox ~ $ sudo apt-get install mysql-server php5-mysql
pi@webbox ~ $ sudo nano /etc/nginx/sites-available/pi
pi@webbox ~ $ cd /etc/nginx/sites-enabled
pi@webbox /etc/nginx/sites-enabled $ sudo ln -s ../sites-available/pi
pi@webbox /etc/nginx/sites-enabled $ cd ~/
pi@webbox ~ $ sudo nano public_html/index.php
pi@webbox ~ $ sudo service nginx restart
```

Craft kickoff

A lot of my work revolved around the Craft content management system. Just like any business, time is money, so myself and the other developers looked at ways to improve EQ's development workflow. There were a few challenges that we needed to address:

Installation and normalisation

Craft installs some example content during installation, but we already understand how to use it and so this content is deleted immediately. There are also some plugins which we use across projects, so these have to be installed. Both of these take time.

Multi-environment configuration

The EQ developers work across three development environments: local, staging and production. Craft has a built in solution to handle multiple environments, however it takes time to configure this for each project.

Asset storage

Images and files in Craft are considered assets and by default, they are stored on the local drive. This works fine until the files are required across multiple environments. Because they've been uploaded through Craft, they aren't under version control and aren't automatically synchronised across environments.

Update Handling

Craft has a one-click updater, which does what it says on the tin. But the time taken to update each installation adds up.

Say it takes two minutes to update one installation, across ten sites and five development environments. That's $2 \text{ minutes} \times 10 \times 5 = 100$ minutes for every update. And there's usually two or three updates a month. That's a lot of time for a simple action.

After experimenting with different solutions, I settled on a new workflow. Along with this workflow was a step-by-step guide on configuring Craft, so no matter who was installing it, each installation would operate in the consistent way. There are some main benefits to this workflow.

Craft kickoff

I created a 'Craft kickoff' installation. This was a production-ready installation of Craft, where the example content had been removed; any plugins we require are installed and multi-environment settings are pre-configured. When a new installation is required, this kickoff repository is cloned, which can then be customised with project-specific settings. This saves a lot of wasted time.

Asset storage

Craft supports Amazon's S3 storage service, so we decided to use that. This means that each file is stored in one location and is available to each and every environment automatically. We're then able to use Amazon's Cloudfront content delivery network to serve those files, improving latencies and speed for all users.

All-for-one updating

The workflow I devised uses a master installation of Craft on each environment. Any subsequent Craft site running on the same server hooks into the master system files, meaning that only one update is required for each environment. Irrespective of the number of websites EQ is hosting, only five installations of Craft need to be updated.

For a bit of humour, these master installations needed a name, and that got us thinking: what is the best thing in the world? Bacon is the best thing in the world. Which God created. Who has been disproven by the Higgs-Boson particle. Thus Higgs > God > Bacon. As a result, the workflow has been named 'The Higgs-Bacon Workflow.'

Summary

I've learnt a lot of valuable knowledge, skills and experience throughout my placement with EQ Design. I'm very thankful for the support and advice they've provided me, so in keeping with their ethos of openness, I'd like to share my thoughts on the overall experience.

Contract

First of all, I'll be transparent and share the details about my workplace contract. I began my placement Tuesday 9th July 2013 and finished it Friday 19th September 2014—just over 14 months. For the first two months, I was employed part-time, working 20 hours per week at £7 per hour. Once I had settled in, my contract was extended to full-time, bringing me in line with the rest of the team.

When my contract was changed to full-time, I was expected to work 33.75 hours per week, again like everyone else (although we're paid for 35.75 hours per week.) At first, my full-time rate of pay remained at £7 per hour. In December 2013, I was awarded a £250 bonus and in January 2014, my rate of pay was changed to a salary of £13,300—equivalent to a 10% pay increase.

My holiday allowance for the year was 24 days (including bank holidays), which I've found to be fairly standard. For my lunch I was allowed 1.25 hours, in addition to as much tea or coffee as I could handle. All in all, the terms of my employment at EQ were very fair, especially for a placement.

Technical skills

The relaxed, friendly atmosphere in EQ allowed me to focus on my work and learning. Given that everyone was so supportive, I wanted repay them and the best way to do that was to produce the best work that I could. I've learnt new skills along the way to help me achieve that.

There's no doubt that my skills improved in every aspect of web development. These are the particular areas where I improved.

Front-end development

Whether he realises it not, EQ's front-end developer Ryan Roberts taught me a lot about the development of front-end code.

Although he rarely taught me directly, his code is very tidy and just by following it through and taking time to understand it, I've learnt useful techniques. In particular, his work has taught me what makes HTML and CSS meaningful. I've now got a much clearer and refined understanding of front-end development.

Back-end development

I've learnt basic unix commands to help me navigate my local machine and remote servers via SSH. I've also developed an understanding of web server software and what in situations a particular programme may be appropriate.

In particular, I've now got a good understanding of the configuration required for a LAMP (Linux, Apache, MySQL and PHP) or LEMP (Linux, Nginx, MySQL and PHP) stack.

Version control

Prior to my placement, I had no experience with version control. My first assignment upon joining the team was to learn about Git, because that was what they used to control code and deployments. By the end of my placement, I could competently use all major features of Git for versioning my code.

Node.js

It was my colleague, Mark McAulay, who introduced me to server-side Javascript and node.js. Being spurred on by his encouragement, I learnt about the main concepts in node.js. I'm now able to develop practical and robust applications using node.js, and push them live to Heroku.

Craft CMS

The start date of my placement coincided with the public release of Craft CMS, give or take a few weeks. Craft fits perfectly with EQ's use-cases, so it became the go-to content management system very quickly. For these two reasons, I've ended up learning Craft in great depth.

Personal skills

Unlike specific technical skills and knowledge, I find that interpersonal skills are much harder to develop. The relaxed working atmosphere and the openness of EQ's staff have really helped me develop as an individual.

Confidence

Confidence is not something that comes overnight, but instead is a combination of knowledge and experience. For me, learning the individual technical skills has certainly helped me because I've now got the knowledge to back up suggestions and ideas.

In the same vein, the time spent working across projects has given me experience. I've found this to be crucial for decision making, as I can draw on similar situations from the past. Now that my placement has drawn to an end, I'm am feeling much more confident in my personal abilities.

Work ethic

My placement has given me my first taste of a full-time job. For starters, it's taught me to be more strict about business administration tasks, such as team communication and time keeping. I find it important to stay focused on a specific job, but as we all know, sometimes attention can drift. I've learnt that to remain focussed, I enjoy breaking away from my usual workspace and clearing my head.

Graphic design

Undertaking my placement at EQ has immersed me in the creative industry and as a result, I've developed a refined understanding of design. In particular, I had a very limited understanding of typefaces before my placement, but now I can identify important details and how they relate to a design or overall brand.

Round-up

Overall, my placement at EQ Design has been incredibly valuable to me. I've learnt a lot about the industry and the tools which they use, and there's no doubt in my mind that my long-term career prospects have improved greatly.

The time I've spent working across the various web-based projects has reassured me that I'm aiming towards the right career, and that studying Computer Science at Robert Gordon University is the right path to take me there. One other thing which I've realised is that I enjoy helping people, whether that's between colleagues, a client or a fellow developer.



"From the day Stuart started with EQ, it became apparent very quickly that he had great ability in a number of areas. His technical skills are second to none and outstrip many a seasoned developer. He has embraced the challenges set to him by our senior developers and was central to EQ's move into using Craft as one of our core CMS frameworks. Additionally he has tackled many difficult development problems for a number of our client websites with great success.

However it would be wrong to just focus on the technical side of Stuart (Baby Stoo as he's become known to us). Stuart is simply a good bloke who is generous with his time and is always on the lookout for ways to help a colleague. A true team-player and a talented one at that."

—**Mark Kemp** Creative Director



Other opportunities

While my placement gave me some great first-hand development experience, it also led to other interesting opportunities. It was my placement with EQ which enabled these, mostly through the colleagues who I worked with.

Northern Lights Conference

The Northern Lights Conference describes itself as 'a one-day enthusiasm booster for geeks of all creeds.' It is hosted in Aberdeen University's King's College and attracts developers from Aberdeen and beyond. EQ Design were a sponsor of the event, and they kindly purchased me a ticket to attend.

There were presentations by nine different speakers on a variety of topics surrounding web development and design, and I found them genuinely interesting. I enjoyed this event because I met new people who were in similar jobs to myself. By listening to the presentations and talking to other participants, I left with a greater understanding of the industry as a whole.

Offset 57

Although this is an initiative started by Robert Gordon University, I may have not attended without the encouragement from my colleagues. The aim of Offset 57 is to bridge the gap between creative students and the industry. To do that, they've hosted various workshops and have held pop-up shops to showcase student work.

The first event I attended was run by the London-based creative collective ShellsuitZombie. Their workshop was run by ShellsuitZombie's two founders and comprised of creative challenges which encouraged us to meet fellow students. It was great fun.

ShellsuitZombie

Andrew Muir Wood and Jonny Burch are the two co-founders of ShellsuitZombie who I met at the Offset 57 evening. We discussed the importance of young web developers in the creative industry and they asked if I was interested in becoming a zombie with them, which sounded like a great opportunity. Since then I've been sharing my experiences of being a young developer through the ShellsuitZombie blog with other projects in the pipeline.

Mobile User Experience

Off of the back of the Offset 57 night, I became aware that ShellsuitZombie co-founder Andrew Muir Wood also co-hosted the Mobile User Experience event in London. After some research, I discovered that they offer a scholarship to attend the two-day, £1,500 event. I subsequently applied and was offered the scholarship.

The event itself discusses cutting-edge technologies and encourages the 100 participants to think in more abstract ways. The speakers presenting at MEX come from all over the world and hold important positions in large, respectable firms. At this first event, I met some influential people, such as the former Creative Director of User Experience at Nokia, a content strategist from Facebook and a user experience designer from Artefact.

Further to this, I was introduced to MEX's founder, Marek Pawlowski. I've been returning to MEX as an event photographer since then, helping to capture the atmosphere from the perspective of a participant. All in all, I've found the forward-thinking speakers to provide inspiration for the direction of my career, as well as what my honours degree project could be focused on.



Craft CMS community

As my knowledge of Craft had developed, I found that I could use it to answer other developers' questions. Pixel & Tonic (Craft's developers) use Google+ and StackExchange as forums for support, and I've enjoyed providing advice and help on it.

"Finally got round to checking Google+ today. @
StuartWhitehead has been killing it! Thanks man! #craftcms"

—Brandon Kelly Founder of Pixel & Tonic and Craft CMS

"Thanks for your help Stu, it is now all working perfectly!"

—Andy Harris Pepper Digital

"Thank you, Stu. This is essentially what I was trying to do. You did it better, and I learned something. Thanks, man!"

—Corey Nicolaides Freelance developer

"+Stuart Whitehead That worked like a charm, thank you!"

—Shanan Galligan Swink

"Cheers mate, I have to switch my habit of defining those dataproviders in the controller (Yii) to defining them in the template. You helped me out of a ton since I now understand that whole system. Thanks a lot!"

—Sjoerd Adema Yii developer

"Wow, thanks Stuart! That's the direction I was heading in, I just wasn't sure if there was a sleeker solution. I suspect I'll have to cache the hell out of this to stop it being a really slow page. Looks good though, thanks again! :)"

—Steve Abraham Craft developer

"Thanks for your help +Stuart Whitehead"

—Jacob Graf Graf Technology

"Cheers Stuart, this was actually one of the first ever things I did on Craft but I never had a practical use for it up until now. Its perfect thanks!"

—Daniel Lightfoot Craft developer

Thanks.

Seriously, thanks. I'd like to acknowledge everyone who helped me out during my placement. Special mentions go to everyone at EQ Design, Andrew Muir Wood and Marek Pawlowski of Mobile User Experience and my friend Ross Gilmore.



References

- AMBROSE, G. and Harris, P., 2007. *The Layout Book*. Worthing: AVA Publishing.
- CONNOLLY, D., 2000. A Little History of the World Wide Web. [online]. Cambridge, MA: World Wide Web Consortium. Available from: <http://www.w3.org/History.html> [Accessed 27 May 2014].
- FLANAGAN, D., 2011. *JavaScript: The Definitive Guide*. 6th ed. Sebastopol, CA: O'Reilly Media.
- KEITH, J., 2010. *HTML5 for Web Designers*. New York, NY: A Book Apart.
- LAWSON, B., 2011. *HTML5 Semantics*. [online]. Freiburg: Smashing Magazine. Available from: <http://www.smashingmagazine.com/2011/11/18/html5-semantics/> [Accessed 28 May 2014].
- NETCRAFT., 2013. PHP just grows & grows. [online]. Bath: Netcraft. Available from: <http://news.netcraft.com/archives/2013/01/31/php-just-grows-grow.html> [Accessed 13 May 2014]
- NETCRAFT., 2014. May 2014 Web Server Survey. [online]. Bath: Netcraft. Available from: <http://news.netcraft.com/archives/2014/05/07/may-2014-web-server-survey.html> [Accessed 13 May 2014]
- PHP.NET., 2014. History of PHP. [online]. Available from: <http://us1.php.net/manual/en/history.php.php> [Accessed 28 May 2014]
- ROBERTS, H., 2013. MindBEMding – getting your head 'round BEM Syntax. [online]. Leeds: CSS Wizardry. Available from: <http://csswizardry.com/2013/01/mindbemding-getting-your-head-round-bem-syntax/> [Accessed 28 May 2014]



