BSc in Software Development

Year 3

COMP07030 Software Design Project

*<TechHub Website>*

*<G00310769>*

*<Claire Finn>*

Contents

[Introduction 3](#_Toc447884727)

[Architecture of the solution 4](#_Toc447884728)

[Class diagram and Data Model 5](#_Toc447884729)

[Technologies used 6](#_Toc447884730)

[Problems Encountered/Solved 7](#_Toc447884731)

[Conclusions 8](#_Toc447884732)

[Recommendations 8](#_Toc447884733)

Student Number: G00310769

Student Name: Claire Finn

Supervisor: Kevin O’Brien

GitHub Link: https://github.com/theultraduffmaster/Main-Proj

# Introduction

*My Project is a group project undertaken by myself and my two classmates Declan Duffy and Thomas McNamara.*

*All three of us feel as if we work well together in college so doing a group project was definitely suited to us. On discussion of our project ideas we all came to the same conclusion; All three of us had a keen interest in the technologies of the future. What will we encounter in the technology sector in even 10 years’ time if technology is this advanced now? We always wondered that so naturally our next question was what language are all 3 of us strongest at that we could incorporate this idea into and work together on? And we all agreed on HTML and CSS as our language of choice. Then we thought the best idea was to do some research on the new technologies and create a website dedicated to this venture.*

*Declan, being good at styling websites told me he would start with that. Thomas, being best at the little website quirks told me he would start with that and try to find little things he could do to make the website more interesting while I went on to do some research on RSS.*

What is RSS? RSS stands for ‘Really Simple Syndication’ and is used to show interesting headlines and stories taken from other websites on your website and these stories are updated regularly. RSS is written in XML (Extensible Markup Language). With RSS you can register your content with companies called “aggregators”. So first you create your RSS document with an “.xml” extension, upload the file to your website and register with the RSS aggregator. Each day this aggregator searches the registered websites for RSS documents and verifies the link and displays details about the feed so you, the client, can link to documents that interest you.

In-Between making progress with RSS, CSS and the website quirks all three of us decided we would each do a “Featured Article” which would be featured on the home page and you can access them at a click. Within my featured article I discuss the history of the Samsung Galaxy S series of Smartphones while adding video links, pictures and more. I also created 2 of the main pages “Household” and “PCs”. I also made pictures on each page stand out more by adding HTML events that zoomed in on the picture as you hover over them.

# Architecture of the solution

This website is one which we realised should have a simplistic look and feel and a uniform design across the entire website. We want people to be able to look at the site and understand it right away so they aren’t confused by the links or extra features. This website should be more about the articles themselves and the confusing features should not be distracting the client from that. This website should allow even the non-technical people (such as the older generation) to visit the site, peruse through the articles at their leisure marvelling at the technology of the future and not be distracted by pop-up boxes, too many links or too much extra content. We modelled this website on sites such as TechRadar with the main links going horizontally across the top for easy access and several featured articles on the main page. The idea of having featured articles on the main page was to give each of us in the group a topic we were most interested in that was most up to date and we could write about it.

On each of the main pages we have pictures and for people who need to take a closer look I , myself, included a HTML event that allows the client to be able to hover over the picture and it will become enlarged, then once the user takes the focus of the cursor off of the picture it returns to its original size. We have embedded videos from YouTube and included surveys for feedback from the client on some of the main pages. We included some buttons with random hand-picked technology facts. The user hovers over the button and a random fact appears. On the main page we have a button, when pressed, allows the user to enter their username. We also have some “cycling photos”, a number of pictures which switch automatically to create a “slideshow effect”.

On each page we have an “RSS Icon”. This is a universal icon used across the web to symbolize that there are RSS feeds within that page. There are links to these RSS pages beside each Icon.

# Class diagram and Data Model

Class diagram

My Team and I decided that it would be best to host our website locally. The reason we chose this is because we felt as if things would run more smoothly. We can test all of our code quickly and efficiently. We spent most of our time expanding the functionality and data included on the webpages so this was the one feature that fell to the wayside.

We had originally planned that we would host our website on Azure server but what with getting everything completed on our project we didn’t get to thoroughly look into this so we left it out.

If we were to do this project again this is the first thing that we would look into.

Main Articles

**Featured**

**ARTICLES**

RSS

RSS

RSS

RSS

RSS

RSS

Other

Gaming

Household

SmartPhones

TVs ss

PCs

Article

Article

RSS

RSS

Article

RSS

Template

(Homepage)

RSS

# Technologies used

*To create this Website I used CSS, HTML, JavaScript and XML. I found that using HTML, JavaScript and CSS was much quicker than using XML as I feel this language is rather laborious and the least thing can go wrong while writing it. It is a very nice language to use but I still feel a little flustered trying to get the little things right. I decided I would look up a tool on the internet that might help me with XML’s design format so I could focus on getting things written correctly. I found a video on YouTube that used a technology called an “RSS builder”. This piece of technology has textboxes allowing you to enter in your data and it automatically generates it to XML for you so you don’t have to worry about getting the format exactly right.*

*Within the RSS Builder it has preset “tags” (textboxes you enter in details to) such as category, description etc. which will automatically render to XML tags. Each time you want to add in a new topic there’s an option for “New Topic”. I saved all of my “Topics” as separate files with an “.xml” extension.*

*On completion of the separate topics I then set about trying to figure out how to incorporate these topics into the website. At first I found an RSS Icon, as stated above, a universal icon used on a webpage to symbolize that there are RSS feeds within that webpage. I placed this icon on every page of the website to insinuate that there were RSS feeds present. I then put a link beside those icons with the .xml file being on the other end of that link. Once the user presses the link it brings up the RSS feed with the articles included.*

*HTML, CSS and JavaScript were the main languages used within this project however. HTML and CSS were the languages I started with in 1st year of college but at the time I didn’t have the appreciation I have for it now. I was being introduced to all of these new languages at once but the one that stuck with me as being the most enjoyable was HTML and CSS. I don’t feel as if I focused on these languages enough since my 1st year so the idea of using them for my 3rd year project really appealed to me. I wanted to get back into writing this language and also expand my knowledge of it.*

*We were notified by our supervisor that Dreamweaver was a good piece of software for getting your website looking uniform and in good format. We appreciated this information but then we thought to ourselves; what if we build this from the ground up? We want to refresh our knowledge of HTML and CSS so why not try doing things ourselves, manually, this will help us to understand the code as it is and get our heads around every element of it. Naturally this took us a lot longer in the long-run, making us miss out on a lot of features we really wanted to include but we feel optimistic that at least the code we have written is ours, written by hand. We had help from the internet of course but mainly our goal was to write the code as much from our own understanding as possible.*

# Problems Encountered/Solved

As I stated above, my main (solo) goal was to get an RSS feed going for our website. I got very far with this and was optimistic that I’d get this feature done although I was disappointed in the end.

The steps were:

1. Get the .xml documents with the articles to be included in the RSS written up.
2. Verify that my XML is written correctly using a “Feed Validation Service”.
3. Include an RSS Icon on the website pages and a link to these .xml documents.

I soon figured out that things were not going to be that easy.

To “validate” my xml code I literally copy and paste it into the textbox included in the Feed Validation Service and press the “Check” button. This service checks the format of your code, syntax and all and gives you feedback.

The feedback I got told me:

“Congratulations, This is a valid RSS feed!”

It then gave me some recommendations:

1. The service told me the general syntax errors which I got fixed.
2. I needed to include a “guid” with each topic to identify it. I had no idea what a “guid” was so after some research I figured out that it was actually a “Global Unique IDentifier”. This allows your articles to have a unique ID attached to it so no duplicates occur. I found out you can use a “Random GUID Generator” to generate random IDs for each article. This fixed the second error.
3. Finally, the only thing that was left was the following:

“ This feed is valid, but interoperability with the widest range of feed readers could be improved by implementing the following recommendations.

[line 38](https://validator.w3.org/feed/check.cgi#l38), column 2: Missing atom:link with rel="self" “

I looked into this and found out that you need to include a link to your own website within the xml code for the RSS feed. My website is hosted locally so it does not yet have a URL. I tried putting in the local “URL” just to test it but the error that came back from the validator was that my path was a “Relative” path. I realise it has to be “Absolute” (include “HTTP”) but we did not manage to host our website on a server so I cannot actually include the RSS feed on my website when it’s local.

# Conclusions

One of the main things I have gleaned from this project is that I should not fixate on trying to get one aspect of the project written up and forgo other more important elements of the project. I found that I was fixating on trying to get the RSS working, going through loads of research and then learning I actually could not get this done until my website was launched on a server. I could have been researching how to get my website onto a server or even other functionality within the project.

I have learned to have more of an appreciation of the HTML language seeing as it is written from the ground up (in a text editor such as Notepad++ or Brackets) where errors are not explicitly pointed out to you. I had to actually research every error that occurred which allowed me to learn from every mistake and know how not to get this error in the future rather than just clicking some “quick-fix” (in software such as “Eclipse” or “Visual Studio”) that you don’t even really think about afterwards as you know that you can just click this “quick-fix” again the next time.

I also learned that working as part of a team was an extremely advantageous thing for me. Every time I had an issue or error I did not go it alone. I simply talked it out with my fellow team members and we would eventually come up with a remedy. The other group members were extremely efficient to work with and this helped me get things done in the long run. It was easy to do out a team plan and delegate certain parts of the project so we could then each focus on two or three things at once each knowing the website was well underway as a result. Our logo for our project is 3 connecting dots with the name “TechHub” written in-between. We designed it this way to symbolise that not only were we three being “connected” as a team through technology, technology actually connects people all over the world.

All in all, I feel optimistic about the work I have undertaken both on my own and in a group. As I said earlier it is not easy or efficient to build code from the ground up but it is a very valuable learning experience for me to be able to look at code and understand it properly in something as simple as a text editor. No time was actually wasted when I am learning from it as a result.

# Recommendations

If I was to do this project again I would try to get my website hosted on a server so I could then incorporate my RSS Feed into it.

I was also thinking I would have loved to have written this project in Asp.net as I could have included a database for dropdown lists and I could have created a “Details-view” which allows the user to “Edit” “Update” and “Delete” items. When I had started this project I hadn’t yet started to learn about ASP.Net in college but after doing a project in ASP I realized I could have been a lot more efficient and included a lot more user-interaction in my website.

I would have added extra functionality such as users being able to log-in successfully (All we have is a button that remembers the user’s name but doesn’t actually save it to a database so it can be remembered overall) ASP could have easily incorporated this as it even has a “Login” functionality in the toolbox. It would be nice to be able to allow the user to be more involved with the website. They might even be notified each time we update the website if we were able to do it again.