

Keep Them Safe: Internet Safety Website

Peter Richards & Jonathan Dempsey

Declaration of Plagiarism

DECLARATION:

"This project was written by me and in my own words, except for quotations from published and unpublished sources which are clearly indicated and acknowledged as such. I am conscious that the incorporation of material from other works or a paraphrase of such material without acknowledgement will be treated as plagiarism, subject to the custom and usage of the subject, according to the University Regulations on Conduct of Examinations. The source of any picture, map or other illustration is also indicated, as is the source, published or unpublished, of any material not resulting from my own experimentation, observation or specimen-collecting." [1]

Student: Peter Richards

Signed

Student: Jonathan Dempsey

Signed

Acknowledgements

We would like to thank Rupert Westrup for all the assistance and guidance he has given us over the duration of the project. We would also like thank all our fellow classmates for the help or wisdom they have offered as well as our families for the support they have given us throughout they year.

You all have our gratitude.

Abstract

This dissertation details the research and development of an Internet security website designed to teach parents/guardians how to protect their children from online dangers. Covering the approach taken towards researching the feasibilities and technologies required for the website, including steps taken towards implementing the end product.

The data for the website is stored in a MySQL database located on a server and is delivered to the front end site using PHP and is displayed using HTML, as well as using a JavaScript library ‘TinyMCE’ to display text editors. The site is designed using Bootstrap which is a HTML, CSS and JavaScript framework for the ease of developing responsive websites.

Testing the code was done by unit testing. Each functional part of the website was developed on its own and implemented one by one to ensure they all worked in the final iteration of the project. Many of the functions were thought of before any code has been written but through user testing added functionality was brought to light. User testing also aided in designing the website, ensuring the site was easy to use for parents/guardians who may not be experienced with computers. Two user tests were done throughout the duration of the project, both using questionnaires.

At the end of the build, the system overall did not provide too many bugs. All bugs were simple to rectify. Usability did prove useful for us to get an understanding of how a user would first interpret the site and allowed us to make small changes to the site which improves the users end experience.

Table of Contents

1. Introduction	6
2. Requirements	7
2.1 Introduction	7
2.2 Similar Product Analysis	7
2.2.1 Introduction	7
2.2.2 Content	7
2.2.3 walkthrough	7
2.3 Requirements	7
2.3.1 User Req Gathering	7
2.3.2 Response	8
2.3.3 Interview Summary	9
2.3.4 User Requirements	9
2.3.5 User Functional Requirements	11
2.3.6 Nonfunctional Requirements	11
2.3.7 Technical Requirements	12
2.3.8 Implementation Platforms	13
2.3.9 Text Editor	14
3. Feasibility	15
3.1 Introduction	15
3.2 Description of proposed system	15
3.2.1 Diagram	15
3.2.2 Server Side	15
3.2.3 Client Side	15
3.3 Issues	15
3.3.1 Server Side Lang	15
3.3.2 Time	16
3.3.3 Testing	16
3.3.4 Front End	16
3.4 Conclusion	16
4. Research	18
4.1 Database Management System	18
4.1.1 History	18
4.1.2 Hierarchical	18
4.1.3 Network	18
4.1.4 Relational	19
4.1.5 SQL	19
4.1.6 PHPMyAdmin	20
4.2 Website	20
4.2.1 PHP	20
4.2.2 HTML	20

4.2.3 Bootstrap	21
4.3 Conclusion	21
5. Design	22
5.1 System Architecture Design	22
5.2 ERD Diagrams	25
5.3 Sequence Diagrams	26
5.4 Front End Design	27
5.4.1 Menus	27
5.4.2 Wireframes	28
5.4.3 Fonts	28
5.4.5 Bootstrap	29
6. Implementation	30
6.1 Back End	30
6.2 Front End	33
6.2.1 Article	33
6.2.2 Video	34
6.2.3 Forum	35
6.2.4 Contact	36
6.2.5 Content Creator Form	36
6.2.6 Admin Functionality	37
6.3 Development Environment	38
6.3.1 PHP	38
6.3.2 SQL	38
6.3.3 HTML	38
6.3.4 Bootstrap	38
7. Testing	39
7.1 Unit Testing	39
7.2 Usability Testing	39
7.3 Test Plan	40
7.4 Result	41
8. Conclusion	42
8.1 Peter Richards	42
8.2 Jonathan Dempsey	43
8.3 Discussion	43
9. References	45
10. Appendices	46

1. Introduction

The goal is to create a website based application that enables parents/guardians to learn about web safety for their children. The website will incorporate the following:

- Video Tutorials
- Articles
- Chat Forums for discussions amongst users
- Video/Tutorial Request Page to request new Videos/Tutorials

The aim is to make it simple for parents/guardians to learn things that can help their children be safe online, for example, tutorials about Facebook's privacy tools. Parent's/Guardian's will be able to register with the site, view tutorials/articles, request new tutorials.

We feel that there is a gap in the market for a product like this. There are websites with some articles and tips and then videos on YouTube. The idea is to create a site that brings these two ideas together in a way easy for parents/guardians who may not be "technologically gifted" to understand the risks out there and also provide them with a way to fix it using video tutorials with step by step guides.

We also feel that there should be interaction between the site and users therefore allowing people to request new videos and tutorials as well as chat with other parents/guardians using a forum perhaps.

The only websites that we could find that provide information are government based websites that do not really teach you about safety but just inform you with links to articles. We feel there is potential for this product as schools including primary schools begin to use computers/iPads in the form of teaching now. We feel it will be easy to market this product through schools. As children are now exposed to the internet from a young age, parents/guardians are becoming more concerned and also more willing to use/learn about these issues.

With the growing functionality of sites that require you to enter personal information, such as a credit card number, it's imperative that users learn when it is or isn't safe to give away details like these as there will always be people trying to gather that information. This could involve Phishing, sending fake emails, tricking users into downloading malicious software or creating fake accounts. These issues have become more common and it will be important for people to learn how to prevent these issues from occurring.

2. Requirements

2.1 Introduction

When creating a project like this it is necessary to identify what will be needed to carry out the full functionality of the end product. Research will be done on sites that carry out some of the functionality that is required to know what to do or what not to do in the project. Identifying the type of users who will be using the site is also important as it will enable insight in how the site will have to be designed. This will be done by interviewing these users to see what they would expect from an Internet security website, as well as know what each of the users would be doing on the site. All the technical requirements will also be identified, and what will be needed to carry them out.

2.2 Similar Products Analysis

2.2.1 Introduction

We Researched and identified a number of websites that had similar aims and providing similar functionality. Webwise.ie and Internetsafety.ie are websites providing users with information regarding Internet safety in the form of articles or videos.

2.2.2 Content

Webwise.ie appears to be the most similar website and best example of what our end project should aspire to, with the addition of more functionality. It's divided up into 2 main sections; one for parents/guardians and one for teachers with subsections including articles and videos. The main page also lists the most popular and most recent articles. It also includes a search function so users can jump to any specific topic should it be available.

2.2.3 Walkthrough

Upon loading the website, parent and teacher sections are listed as well as a number of recent articles to click into. When loaded into one of those headings, a list of articles are displayed relevant to the heading. Clicking on an article link will display the article, as well as related articles, categories and tags. The site does not allow users to interact with one another which this project will hope to achieve.

2.3 Requirements

2.3.1 User Requirements Gathering

To help create the user requirements interviews were conducted to gather as many possible features that users would like to see on the site. The following questions were asked:

1. Would you engage in online forum chats about Internet security with other parents?
2. Would it be easier to learn from a video or an article?
3. Would you register on a site to contribute your Internet experience for child safety?
4. Would you help other users with problems that you feel you'd be able to help with?
5. Are there any features you'd like to see on the site?

2.3.2 Responses

Person 1

1. Not an active user on most sites, unlikely
2. Video
3. Not an active user on most sites, unlikely
4. Yes
5. Most Viewed/Liked search feature.

Person 2

1. Yes
2. Video
3. Definitely
4. If I think I can help I would
5. Rate videos

Person 3

1. Yes
2. Article (I wouldn't like stopping and starting videos)
3. Yes
4. Yes
5. Giving the video timestamps to skip to

Person 4

1. Yes, I find that our schools do not do enough at the present moment
2. Both. I like to read but also tutorials that you can see exactly how to do things would be helpful
3. Yes
4. Yes
5. I would like to see a forum of some sort so parents can post questions

Person 5

1. Yes. It is hard to find good information on the internet at the moment
2. I would prefer to listen to someone rather than read an article
3. Yes
4. Absolutely
5. I would like to see an easy to use catalogue of videos that are divided into topics such as Facebook, Dealing with Adult Content, Illegal websites.

2.3.3 Interview Summary

Through conducting the 5 Interviews we found that 4 out of 5 interviewees would be willing to register and participate in helping others regarding internet security. In addition, the majority of the users would prefer video or a combination of video accompanied by an article. The extra features they would like to see added generally include features that larger websites already have implemented such as a video rating system to show what content is of good quality, video timestamps to skip to important parts of the video, the ability to search content based on popularity and topic, and finally the ability to interact with other users through a forum.

2.3.4 User Requirements

The application will have three main users:

- Administrator
- Content Creator
- Parents / Guardians

The Administrator is in charge of the overall website and has the highest level of controls that the application has to offer. The administrator will have access to all these features as well as the systems code. The administrator will be able to make changes and updates to software and code as required to keep up with the latest technology. The administrator will also be able to add/remove/edit accounts manually.

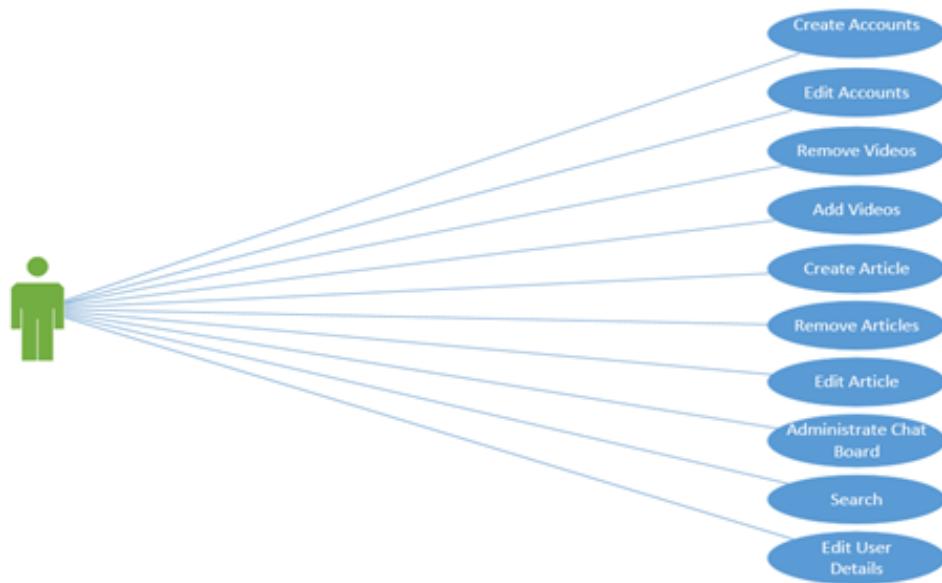


Figure 1, Admin Use Case Diagram

The Content Creator will have a more basic approach to the site. They will be only able to control content. The Content Creator will have a few basic rights in administration as an

administrator cannot be available 24/7 so they will be able to administer/regulate any chat boards to help deal with any possible unwanted posts. On top of this, they will also be able to create content for the site such as articles and videos. The content creator will act as a mediator if required between standard users.



Figure 2, Content Creator Use Case Diagram

The next user group is going to be the most commonly used with the software. This user group is used by parents/guardians or other people looking to learn about internet safety for children. This user group will be able to perform basic operations such as search the site, view videos and interact with other parents/guardians through the chat boards. The group only has basic rights so they cannot access/edit content except in the chat board.

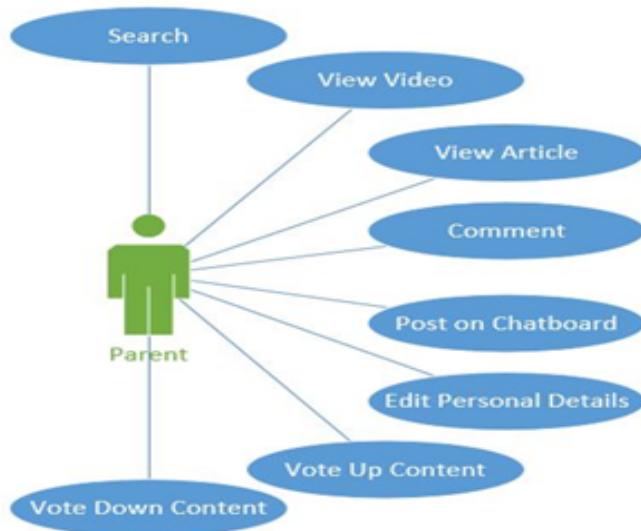


Figure 3, Parent Use Case Diagram

2.3.5 User Functional Requirements

When a user loads the site, the software will first load them onto the main page. The software will give them a basic view of the site until they login. Once the user logs in the software will then become more active as features of the site become unlocked such as viewing videos and accessing articles as well as access to chat boards. The website once logged in, will provide the main functionality to the user as they have access to these features.

The content creator will have extra facilities. These will include file/video uploads, article creating facilities and also access to the chat board moderator functions. The site will allow content creators to upload their videos which will then be made publicly available to parents/guardians or other users. The site will also allow creators to enter articles which will be stored into a database.

The site will allow the administrator have total control over content and chat boards. The administrator will be able to create new topics and add new pages to the website. On top of this, the site will allow the administrator to manage accounts. The administrator will also have the privileges of a Content Creator and also the basic User.

2.3.6 Non-Functional

Non-functional requirements the project will focus on are security protocols, maintainability and usability.

Security

Security protocols will be put in place to protect the Users personal information by having strict password requirements using SQL. The password requirements would need the user to pick on a password that's long, is case sensitive and has to have a numeric character. Security protocols will also be used to protect the database server.

Usability

Usability is important as less experience users will have to be able to find what they are looking for and find it easy to post their own comments/videos. To ensure this the sites navigation will be clear and easy to use as well as clearly labelled actions such as posting/editing/removing their own content.

Maintainability

The maintainability and scalability of the database won't be as essential at the beginning of the projects life, but as time goes on will become an important factor.

2.3.7 Technical Requirements

For the most part, data will be stored in a database (MySQL/phpMyAdmin). The only data that will be stored outside the database are the video files. Things such as articles and conversation topics will all be stored as entries in database tables. Each sub system will have its own table in the database.

The user interface will be kept as simple as possible due to the audience that the website is trying to capture. As the users who potentially are going to be accessing the site will not be tech savvy, navigating the site should be kept as simple as possible. The majority will be done with buttons and event listeners which will trigger functions. Below, when a user clicks on a one of the topics, it will then load a new page with the topic in full view with expanded details. The majority of the site will work like this by clicking on something; it will bring you to a new page with more details on what you clicked on.

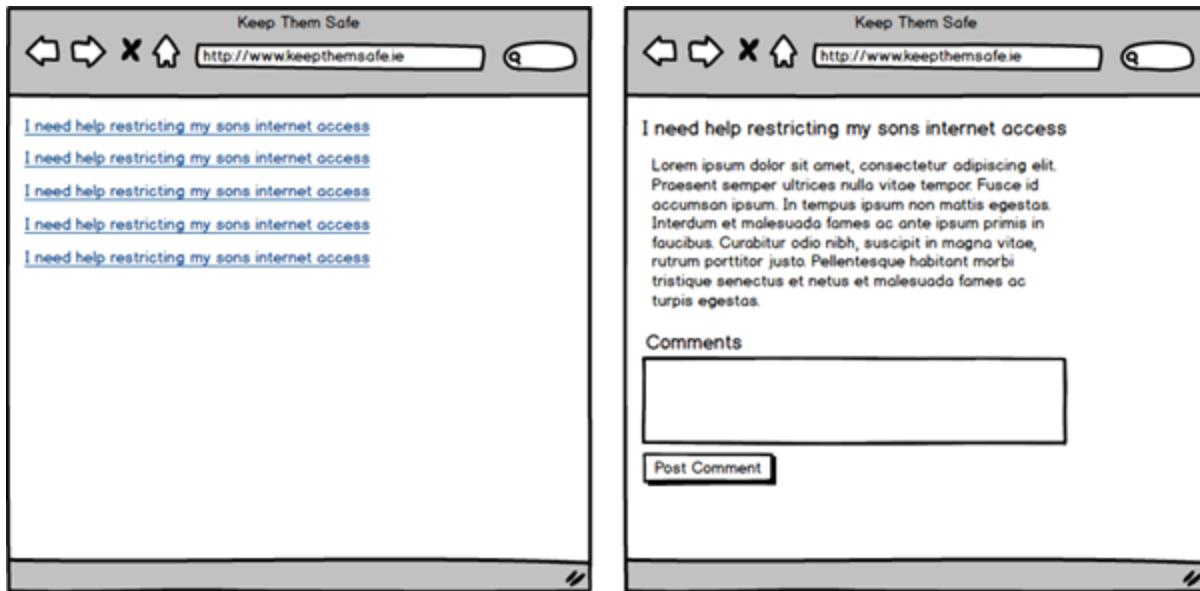


Figure 4, Wireframe Article Navigation

The way that the system will process user events will be a client – server – client system. When a user for arguments sake requests a list of discussion topics, the request is sent to apache on the server which will fetch the data from the database table that belongs to discussion topics. A sql request will retrieve these and php will then format them correctly to be displayed on the page for the user to view.

2.3.8 Implementation Platform

The technical requirements outlined above will be implemented using a combination of PHP, SQL and HTML working in relation with a database. The project members have some experience working with all these technologies though have only worked with them on a much smaller scale than this project. Using these existing skills will assist greatly in helping to understand how to implement the full functionality of the project. However, what has already been learned will only act as the foundation for the additional skills will have to be learned.

From previous work with the login functionality, it can be ensured that the user can login using an email and use validation to help with security. Research til be done to expand on the knowledge to be able to allow multiple types of users all with different levels of access to the websites functions. The comment, article and forum systems will require the use of current knowledge of database tables in a new way. This will include the ability to add, remove and edit database elements from a HTML page that only a specific type of user will be able to use.

The video upload uses PHP and JavaScript functions which will allow a user to upload a video file to the database that is then to be displayed in a HTML page. The current implementation of this allows the user to upload any sort of file which will need to be fixed. Although both project members have experience with much of the functionality of the projects functionality, additional research will have to be done in all technical requirements to ensure the ability to

see the project through to the end. This can be done by learning through tutorials, finding books focusing on these requirements and looking at examples where these functions are already put into practice.

2.3.9 Text Editor

As content creators are going to have to submit articles in some form, it was decided to have users write it out into a text editor on the website rather than load in an outside document. To accomplish this TinyMCE will be used. "*TinyMCE is a platform independent web based Javascript HTML WYSIWYG editor control released as Open Source under LGPL.*" [2]

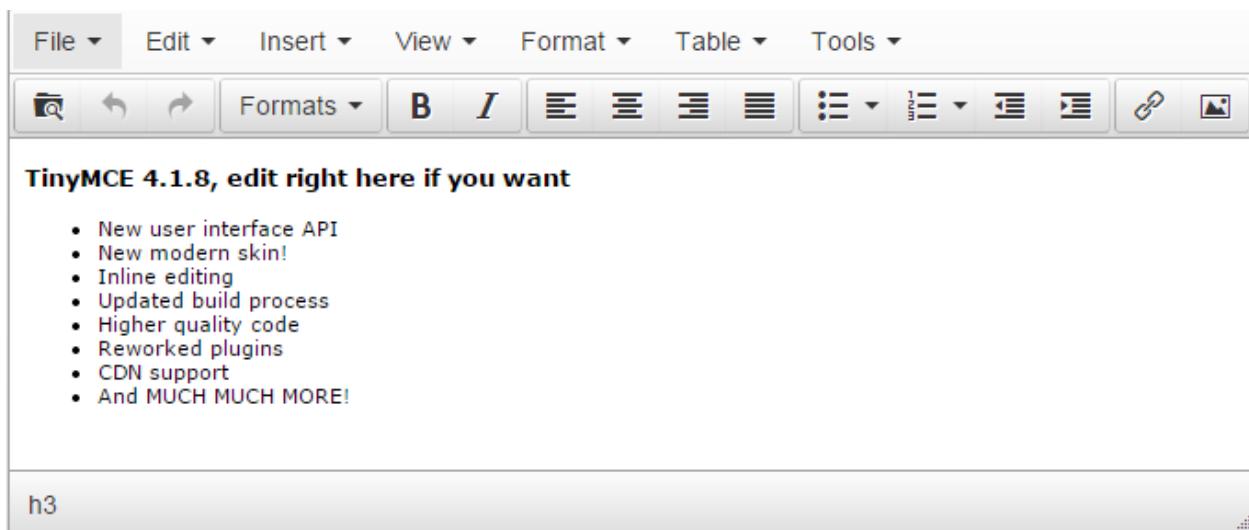


Figure 5, TinyMCE Text Editor

3. Feasibility

3.1 Introduction

This section of the document focuses on the feasibility of the project. A feasibility study was carried out to determine any risks involved with the project. This was done by looking at what we had to do and determining any risks that could cause problems during the implementation of the software.

3.2 Description of Proposed System

3.2.1 Diagram

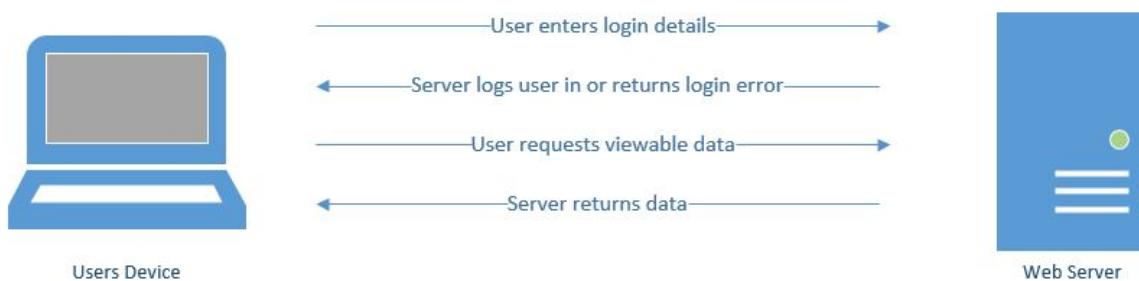


Figure 6, System Model

3.2.2 Server Side

The KeepThemSafe website will be managing all the data that gets stored on the database to be displayed on the site or used for other functionality. The user will request information from the server, like articles and videos, the server then returns the data back to the user. The same steps are taken for adding, updating and deleting data. To interact with the database MySQL is used on the server side which is run on phpMyAdmin which “*is a free and open source tool written in PHP intended to handle the administration of MySQL with the use of a web browser.*” [3]

3.2.3 Client Side

The site is written in HTML and a CSS tool called Bootstrap. The functionality is achieved using PHP and SQL to pull data from the database.

3.3 Issues

3.3.1 Server Side Language

This project requires a greater understanding of PHP. Although limited knowledge of PHP is a help, most of the advanced features of the site will be a learning experience such as video uploading. This is something new to be learnt and requires a stronger knowledge of PHP.

After researching more into this area, it has become clear that the task at hand is not impossible but will require a small bit of learning to expand the skills set currently at hand.

With the use of temporary PHP folders, creating the scripts to move these files will be the most challenging part as when a video is uploaded, it goes to a temporary holding folder and must be moved and the location stored in a database.

3.3.2 Time

This is going to play a major part in this project. After creating this document, it is understood the full task at hand and the length of work that needs to be completed in order to complete the project. The longest part of this project to implement will be the uploading of files. However, this can be done before the design of the software takes place and can be simply adapted to fit the design.

3.3.3 Testing

To test the application, the more difficult software is going to be created first such as

- Creating /Editing Articles
- Uploading Videos
- User accounts and privileges

Once the more difficult stages as above are completed, testing will be conducted to make sure that they are fully functional and ready to be implemented into the main body of the website.

After each section of software is created, it will then be tested to make sure it is functional. This will hopefully reduce time in the long run as to not spend days fixing problems after the entire site has been built but rather with each section.

Ideally we will follow the Waterfall model however during the implementation stage; it is felt the best method for code implementation will be to create a system e.g. upload a video, then test it on its own. Once it passes, it can then be integrated into the final product.

After the entire application is complete, it will be made available to a select few to test and report any bugs. Ideally, the students would want to solve most of the bugs before beta testing and to allow user testing as a final phase of testing to fix small minor issues that the ordinary user may have such as bad navigation or unclear processes that the user may encounter.

3.3.4 Front End

To present the best user experience we decided to use CSS minifier to ensure that the CSS files do not cause too much of a slowdown on the site. When tested on a sample code of CSS, minifier cut the file size down by half.

3.4 Conclusion

To conclude, the website will be written in HTML and PHP. It will be designed using Bootstrap so we can pump out prototype CSS code at a fast rate. Data will be stored on a database and can be accessed and managed using PHP and SQL.

Based on the following architecture, we have a working HTML front page to the website, designed using Bootstrap. We also have a basic login system working as well as a simple adding article and adding video system. With these results, this architecture will continue to be used for the remainder of the project.

4. Research

4.1 Database Management System

4.1.1 History

A Database Management System (DBMS) is used to create, maintain and manipulate the data it stores. A DBMS is a computer software that manages databases and can be implemented using an array of different models like the relational, network or hierarchical DBMS.

4.1.2 Hierarchical

In 1966, IMS was released and was the first commercially available DBMS. It was developed by IBM, originally intended to be used for the Apollo program and used the hierarchical model. This model was based on binary trees, which means relations were represented in a 'parent-child' format where each child has one parent but one parent can have many children. [5][6]

There were many benefits to using hierarchical DBMS such as security, integrity and data independency. This led to easy, efficient search results. Despite the advantages, Hierarchical DBMS was a struggle to implement and without proper standards, made it complex and difficult to manage many relationships. [6]

Hierarchical database model

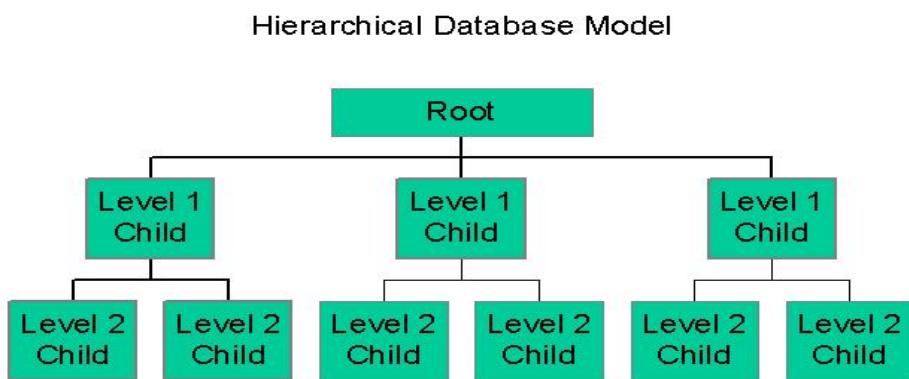


Figure 7, Hierarchical Database Model, [4]

4.1.3 Network

In 1964, Charles Bachmann developed the Integrated Data Store (IDS) which was an early network database mostly used by industry and commended for its high performance. A group called CODASYL had the system standardized. This was a consortium was to guide the

development of a standard programming language that can be used on multiple computers. In the network model, each child could have multiple parents, in oppose to the single parent in the hierarchical model. It is composed of a set of relationships, these sets represent a one to many relation between parent and child. Faced with the same issues as the hierarchical DBMS, network DBMS was complex, difficult to maintain and design, mainly caused by a lack of structural independence. [7][8]

4.1.4 Relational

Edgar Codd, an employee at IBM, was not happy with the network/navigational model of the CODASYL APPROACH. In 1970, Edgar proposed the Relational DBMS. This DBMS was to be used to enter data and work well with large databases that used tables. [10]

These tables would be linked together as a one to one, one to many or many to many. This meant that if irrelevant data was taking up space, it was simple to remove the unwanted table along with all the other entries in other tables linked to the record were removed too. Relational DBMS builds a new table based on the required information from existing table, using the relationship of similar data to increase the databases speed and versatility. [7][10]

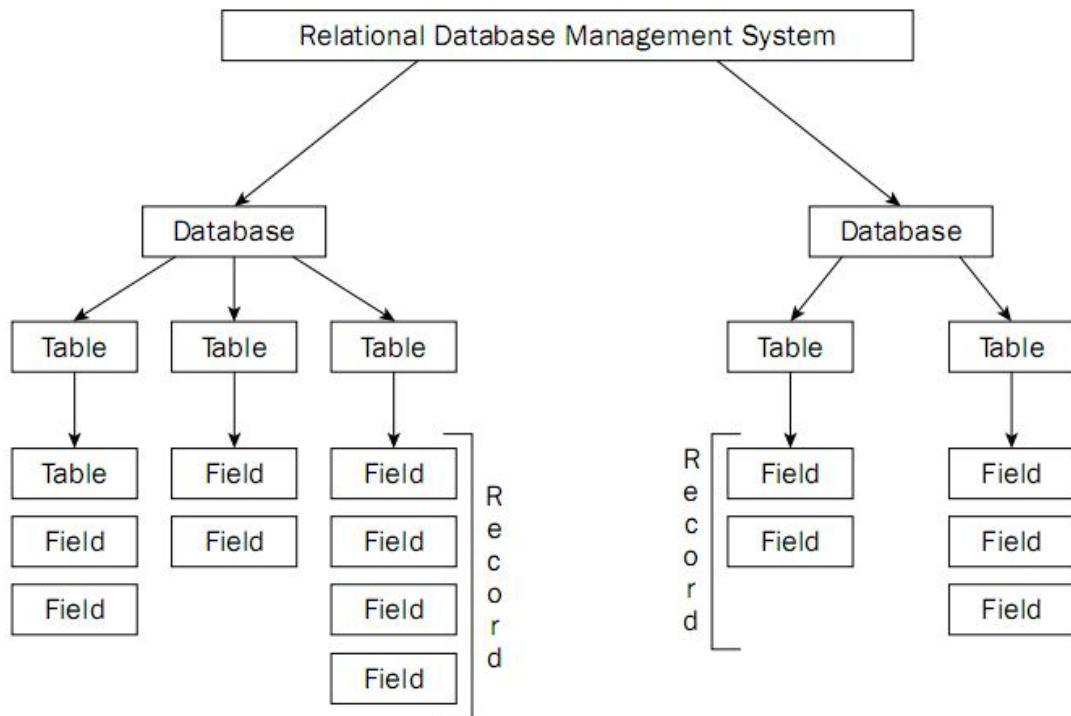


Figure 8, Relational Database Management System, [9]

4.1.5 SQL

Before the release of the first Relational DBMS, each database had its own unique programming language and physical structure that the programmer had to be able to understand. Now we have SQL (Structured Query Language), which is a standard language

used to communicate with a Relational database. Edgar Codd of IBM developed the first prototype of SQL using his own paper “A Relational Model of Data for Large Shared Data Banks” as a model to base his work off. In 1979, the first SQL product, ORACLE, was released soon after Edgar’s prototype, and is a distinguished leader in relational database technologies today. [10]

SQL’s purpose is to provide an interface to a relation database and all SQL statements are used as instructions for the database. SQL provides the following features for its users such as automatic navigation to the data, process sets of data as groups rather than individual units and uses statements that work well on their own. These statements perform a number of tasks, mainly CRUD (Create, Read, Update, Delete), Querying data and controlling database integrity and consistency. [11]

SQL is supported by most relational database management systems allowing the user to transfer their skills from a number of DBMS. In addition, SQL is portable, which means it can be moved from one database to another with little effort and modifying. [11]

4.1.6 PHPMyAdmin

phpMyAdmin is a tool created by Tobias Ratschiller in 1998, that is written in PHP with the intention of handling the administration of MySQL servers. It is capable of creating and deleting databases, creates/deletes/updates table and views/deletes/edits/add fields, manage keys on fields and privileges, manage triggers and stored procedures, execute any SQL statement and can export data into various formats. To run phpMyAdmin, a web server is required to install phpMyAdmin’s files as its interface is based off your internet browser. [14]

4.2 Websites

4.2.1 PHP

PHP is an HTML embedded scripting language used primarily in web applications. Rasmus Lerdorf created PHP in 1994 and was first written in C. Rasmus had created PHP originally to track visits to his online resume, naming the script Personal Home Page Tools, mostly referred to as PHP Tools. More functionality was added by Rasmus providing more uses and implementations, including database interaction and a framework where users could develop simple dynamic web applications. In 1995, the source code for PHP was released allowing programmers to improve the code and fix bugs. [12]/[13]

PHP pages contain HTML with embedded code that performs a function. It uses a start and end processing instructions that allow the user to jump in and out of PHP by using ‘`<?php`’ to start and ‘`?>`’ to end. PHP is focused mostly on server-side scripting but can also be used in command line scripting and writing desktop applications. It was can used on all major OS’s and is supported by most web servers. [12]

4.2.2 HTML

HyperText Markup Language (HTML) is the standard markup language used to compose web pages. HTML is still an evolving language, still going through versions with added functionality, from HTML 1.0 in 1991 to HTML 5.0 in 2011.

When HTML 1.0 was released not many people were involved in web creation and the language itself was extremely limiting, using only around 18 basic elements. When HTML 2.0 was released it was made the standard for web creation up until 1997. Its properties and elements defined the basic working abilities for most current browsers and is seen as a benchmark for when web design came to popularity. [16][17]

After HTML 2.0 had been made the standard and the complexity of web content increased, more elements and tags were needed. HTML 3.0 draft was introduced and included much of the new elements and functionality user desired. However, web browsers were slow in implementing these elements, and even then very few were added causing HTML 3.0 to be dropped. [17]

In 1996 at the World Wide Web conference in Paris, HTML 3.2 (code named “Wilbur”) was reviewed. It was a combination of features from HTML 2.0, extensions created by web vendors as well as some surviving features of HTML 3.0. As 3.0 had failed and 2.0 becoming redundant, 3.2 quickly became the new standard for browser and web creators before it was even recommended. It is still supported by most browsers today. [16][17]

HTML 4.0 (code named “Cougar”) took most of its new features from the failed 3.0, trimmed old tags and focused on supporting cascading style sheets (CSS). 4.0 was recommended W3C in December of 1997 and made the standard in 1998. [16][17]

After 4.0, programmers focused on a proposal for XHTML2. However XHTML2 wasn’t catching on as it began to seem unrealistic and dull. It was decided by W3C that XHTML2 should be discontinued and everyone to focus on HTML5 to become the new specification. HTML5 is designed for the web, for now and in the future, and its development process has been relatively slow but considered. [16][17]

4.2.3 Bootstrap

Bootstrap was created by a developer at Twitter in 2010 and since then has become a popular front end framework and open source project. [15]

Bootstrap was presented at Twitters first ‘Hack Week’ and was received extremely well by developers of all skill levels after only a few months into Bootstraps development. It was then released on August 19, 2011 had since then, has had over twenty releases including two major overhauls with version 2 and version 3. Version 2 saw the introduction of responsive

functionality while version 3 had its library rewritten to support its responsive functionality to be a mobile first framework.[15]

4.3 Conclusion

The results of researching technologies and their history helped give a better perspective on the project as well as confirm that the technologies chosen to apply to the project were the right choice and justified.

5. Design

5.1 System Architecture Design

This section will look at the application in a more technical way and will look at the technology required to run the application successfully. The system will be created to help parents/guardians to gain knowledge on internet safety for their children accessing the internet and what possible dangers there are. The system will also feature content to explain to parents/guardians how to secure children's computers/tablets and how to keep safe on major sites such as Facebook, YouTube and other sites. The main idea is to create content that parents/guardians can follow i.e. a step by step video on how to secure a Facebook account or how to deal with any content that may be unwanted.

The previous document looked at the requirements for the user and discussed the user requirements for the application. With three main users (Administrator, Content Creator and the Parent/Guardian) it was decided how each user would interact with the system and the functions that they would be able to perform.

The parent/guardian is the priority user and will have the most accounts. The content creator and administrator will be limited in terms of the number of accounts.

The system will be of a request and response type system. The user will request something from the site and the site will return the request. For example, a user logging in:

- User requests site from server
- Server sends copy to user
- User enters details and logs in
- Server checks details in database and starts session or returns error

On the server side, the main system will run off apache with help from php, mysql and server storage due to content file size. As videos are being used stored on the server, there is a need for large storage space as a video file could easily reach 1GB each. As this is the case, the testing server has 450GB of free space for content. The database will have a number of tables for things such as articles, video links, users and more.

Article Database setup

A table will be created on the database using PHPMyAdmin. A table called article is created with the heading articleId, articleTitle and articleContent. Within the project folder, a folder to contain all of the articles functionality is set up. First, 3 classes are created: Article.php, ArticleTable.php and DB.php. These classes act as the foundation which will allow the ability to connect to the database and access the appropriate table/content. The next step is to increment the functions.

Article View

To allow the articles to be viewed, an HTML table is made that will link to the content of the database and this table expands based on the number of articleId's found on the database. Clicking on the view button will take the user to a new page where another table simply grabs content and title using get methods.

Article Add

Next the add functionality is incremented. Clicking on add article will take the user to a new page with an add article form. Filling in the details of this form will add a title and content directly to the database. Add_article_form.php is also linked to add_article.php which manages the forms validation.

Article Update

The update article function is then added. This works almost identical to the add function with the same sort of file layout except instead of adding to the database, it takes data already on the database and allows the user to alter the content or title.

Article Delete

The final increment adds a delete function. This simply locates an article based off its Id. If the articles Id is found, a delete function runs which removes from the database.

User Accounts Database setup

A table called users is created in the database with the headings id, email, password and role. Same as article, 3 classes are created that will allow the website to access the database and access the proper table/content. These classes are called User.php, UserTable.php and DB.php.

User Accounts login

For the user to log in they will have to be able to query the database for their username and password. To do this, a login form is required to handle the HTML form and a class is needed

to manage the logins validation. The login form will be used to check if the username and password entered into the text field exist on the database and that they go together. If they exist together, the user is logged in, if not, an error is returned and the user is not logged in. The validation file checks for a two conditions, it checks if the text fields are empty and if they valid on the database.

User Accounts Register

Registration works similarly to the login function except instead of checking if the data exists, it adds to the database. Two files are created to manage the HTML form and to handle the validation. The registration form has the username and password text field in addition to a confirm password field. Before the data is submitted, the validation file checks for 4 main conditions: If the username is an email address, if password and confirm password match, if the text fields are not empty and if the username entered isn't already registered to the database.

Upload Video System

The upload system uses a php to move the file to permanent storage and add a link to a database table to store something similar to a catalogue of videos and their exact link location. Initially the video file is uploaded into a temporary php storage folder that php uses. After this, a function will move the file into a permanent location and that location will be stored in the database along with a name, date and sub folder.

To view the videos, a list will be populated from the database table and displayed. When the user clicks on one of the video links, a new page will be opened which will display a title and the video which users can watch.

Topic System

The topic system will allow users to ask questions for other users to answer. The system will be anonymous. The user will be able to view a list of topics that have been created by other users and on clicking them, will open a new page and show in more detail the title, content and also be able to comment on it. This will all be stored in a database table.

5.2 ERD Diagrams

As the forum is anonymous we do not associate it to any user accounts. Each sub system has a one to many relationship. The most complicated system that just deals with the database will be the discussion system. Each topic will contain a title, content and also multiple comments.

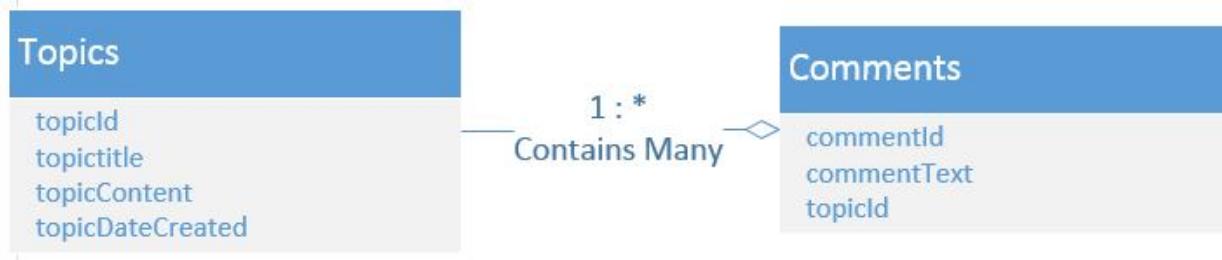


Figure 9, Topics ERD

5.3 Sequence Diagrams

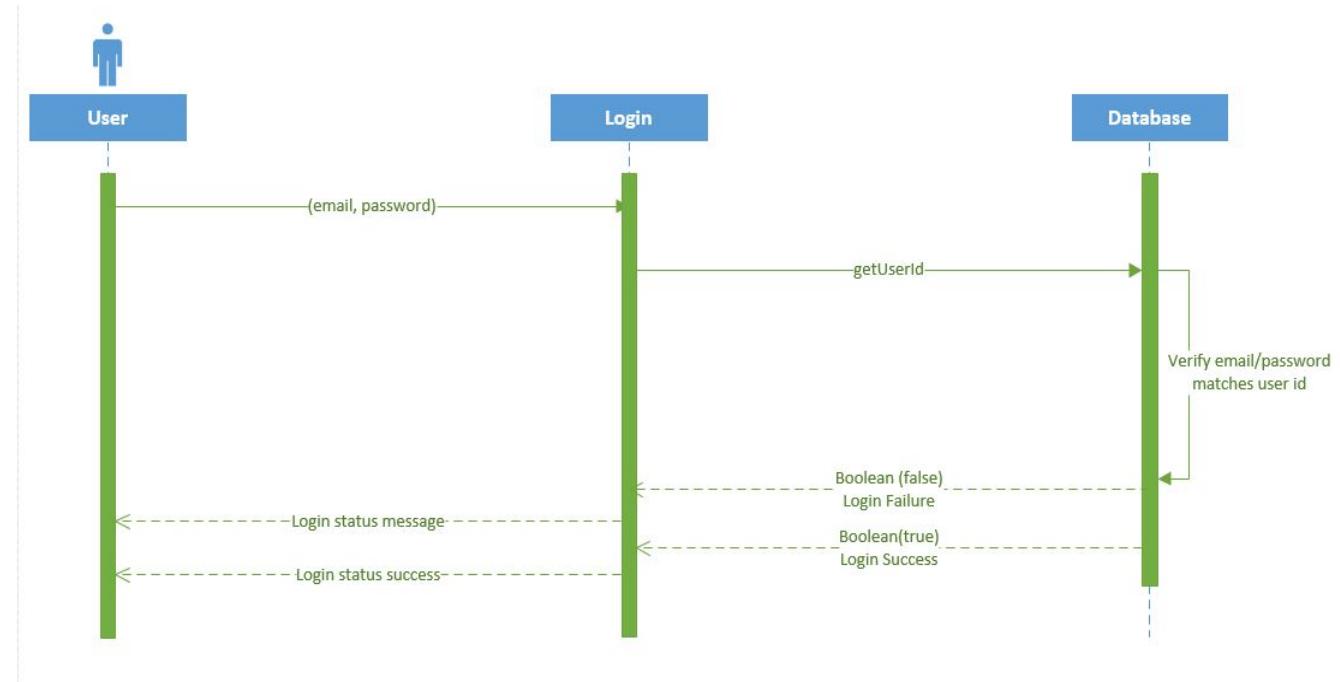


Figure 10, Login Sequence Diagram

The sequence diagram above shows the process our system takes to ensure a user gets logged in successfully or declined. The user enters their email and password into the fields and click the login button. Once the login is clicked a request to acquire a user id is made. The system verifies that the email and password entered match an existing ID number. The login either fails because the email/password don't match a given ID and a login failure

message is sent to the user or the email/password matches an ID and the login is a success. A final function not shown in the sequence diagram is that if the users access level is admin status, an additional button displays on the screen that allows the admin to perform actions normal uses are not allowed to perform.

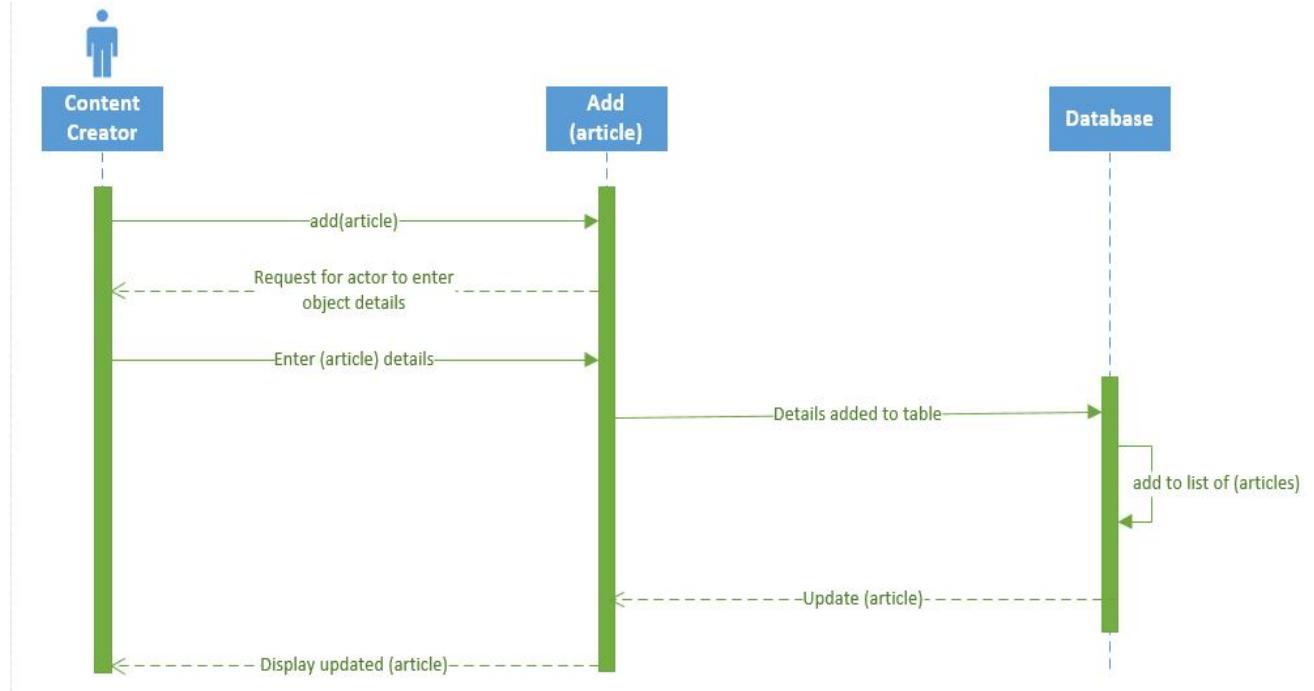


Figure 11, Add Article Sequence Diagram

In this second sequence diagram we see how a content creator can add content to the website. In this example, the content creator will be adding a new article, although the user is adding an article the same principles apply to adding a video and posting to the forms. The user first clicks on the add article button to which the add function requests the user add the articles details. Once the user has added the articles details and submitted them, a new id in the database is added with the associated article along with the articles own detail columns (title, content). The updated database table then gets updated on the website for the users viewing.

The two sequence diagrams shown above outline the functions of the login and add function for articles. However, as the majority of the sites functionality follows extremely similar steps, it was deemed unnecessary to do out a sequence diagram for each one.

5.4 Front End Design

5.4.1 Menus



Figure 12, Main Navigation

The site uses a main navigation bar which is located at the top of the page. When a user logs into the system, the navigation bar will change depending on their account type. The parent/guardian user group will show the following basic nav. The user will need to be logged in to the system to be able to access Articles, Videos and the forum. We do this to keep an element of privacy to the site for users.

The content creator's navigation bar will also include another option called Create. This will act as a drop down bringing the content creator to access a menu to select either create an article or upload a video.

The admins navigation bar will include an extra drop down menu called Admin. Here there will be features to access the approval list, members list, remove/edit articles and videos.

Through a privileges system to log in users, the system will be able to recognise the account type and choose the navigation bar to suit accordingly to the user's privilege level on their account.

At present, when scrolling down the page, the navigation bar is lost. There is a possibility to change this to a navigation bar that will scroll with the page and remain at the top allowing the user to always have access to a method to change page as well as a step back function to return to the previous page.

5.4.2 Wireframes

The general layout of the header and footer on the webpages are consistent. This means that when working with wireframes when both the header and footer are finished it's just a matter of using those templates and adding a new body for a page when needed. When deciding on a final colour design for the website, it was clear that the site had to have a clean and inviting colour scheme that would ensure that users could easily distinguish what were clickable links

and sections clearly labelled. As the site focuses on most likely inexperienced users, these details were crucial. A number of colour schemes were experimented with to help in deciding what looked most reliable.

These wireframes are located in the Appendices at Appendix 1, 2, 3 and 4.

5.4.3 Fonts

Just as the colour schemes, the same process of deciding a font followed similar procedure. A desired font was chosen primarily on the basis that the font could be easily read by the users. The font that currently appears on the page is likely to be used but should that change, there are a number of backup fonts ready to be used. Most of the backup fonts were found using google fonts.

Examples of these fonts are found in the Appendices at Appendix 5, 6 and 7.

5.4.5 Bootstrap

Bootstrap is a css/layout framework that is designed for rapid prototyping. Bootstrap allows a user to take their pre-built css and quickly apply it. The framework mainly focuses on CSS however they do use small bits of javascript too. The advantage of using such framework means that the user has to worry less about a design that will work on multiple platforms and allow more time to be spent on the coding rather than the layout/design of the website.

Bootstrap comes with a few different css files. The main one which comes with a readable format and the other as a .min version. The same applies for the javascript files. Using the .min files will allow for a quicker load time for the user.

Bootstrap implements a simple grid system similar to the 960 grid. They have cleverly designed this grid system to allow for media queries to create a responsive website. Bootstrap also implements their own design of things such as

- Typography
- Tables
- Forms
- Buttons
- Images
- Helper Classes
- Navs
- Panels

Having these pre-made allows the user to create a site extremely quickly as fiddling with css can take up valuable time.

In the Appendices is a sample of the grid system that they use for mobile and desktop layouts at Appendix 8 and 9.

To implement a layout or style from bootstrap you need to simply use class="layout type" in the button tag. Bootstrap follows the same way throughout all html code. Simply adding the class you wish to use will call the style from the CSS file where bootstraps CSS is stored.

6. Implementation

6.1 Back End

To start creating the system, first the server that would run the website would need to be set up. Taking an old computer with the following spec, I was able to create a stable working environment using XAMPP software. The server is being kept in an attic which during the winter is a very cold environment which would be ideal to keep the system temperature down.

Computer Specifications

Processor: Intel i7 930 @ 2.80GHZ

RAM: DDR3 12GB @ 1600MHZ

PSU: 650W

Motherboard: ASUS P6T SE

Graphics Card: NVIDIA 8600 GTS

Hard Drives: 1 - OS, 2 - Dedicated to XAMPP, 3 - Home Storage Cloud

Internet Connection - CAT5e direct to modem with 200 mbps down and 20 mbps up with UPC



Figure 12, Server Box

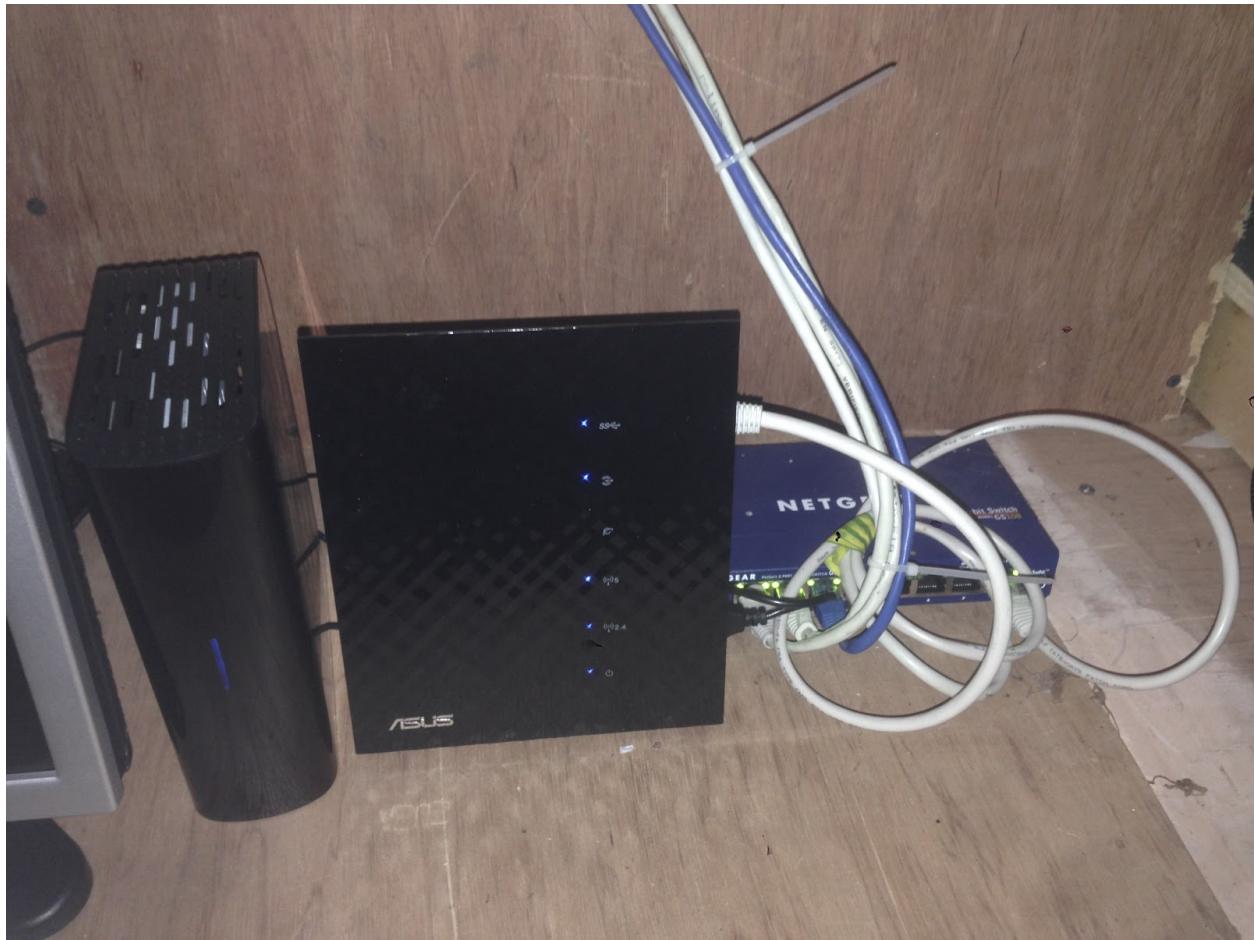


Figure 13, Network HD - Holds backups of the web server HD, taken daily

After re-building the PC and cleaning all the components, a fresh install of Windows 7 Ultimate 64bit was installed. 64 bit was chosen to allow all of the available RAM be used instead of the limited 4GB with windows 32 bit operating systems. The OS was installed onto a dedicated HD to allow for a quicker boot time and also system response time. This then left us with 2 HD's. One was going to be dedicated to running XAMPP and another was to be used as a file cloud server.

The first software to be loaded onto the server was an antivirus - Avast. This would protect the system to any files that may be uploaded as the software is designed to deal with them automatically rather than giving the user an option. This proved a good idea as the system was going to be left unattended most of the time.

XAMPP was then installed onto its dedicated HD. In this case, we chose a 400GB HD to allow for many video files. As the video files would range between 50MB and 400MB (estimated), the larger HD would allow for plenty of videos for testing. After the installation of XAMPP, some modifications to PHP and also Port Forwarding to the server would have to be done.

Firstly, the temporary php file size had to be changed to allow for the large video files to be uploaded.

```
766 ; Maximum size of [POST] data that PHP will accept.  
767 ; Its value may be 0 to disable the limit. It is ignored if [POST] data reading  
768 ; is disabled through enable_post_data_reading.  
769 ; http://php.net/post-max-size  
770 post_max_size=2000M
```

Figure 14, Code Snippet

Secondly ports had to be forwarded in the router to allow external access to the server computer. Port 80 for Web, Port 25 for FTP and and a few others also for other applications that were going to run on the server. In total, 10 ports were opened to the server.

9987	to	9987	192.168.1.21	9987	to	9987	Both ▼	<input checked="" type="checkbox"/>
80	to	80	192.168.1.21	80	to	80	Both ▼	<input checked="" type="checkbox"/>
3306	to	3306	192.168.1.21	3306	to	3306	Both ▼	<input checked="" type="checkbox"/>
21	to	21	192.168.1.21	21	to	21	Both ▼	<input checked="" type="checkbox"/>
32400	to	32400	192.168.1.21	32400	to	32400	Both ▼	<input checked="" type="checkbox"/>
<hr/>								
1000	to	1000	192.168.1.21	1000	to	1000	Both ▼	<input checked="" type="checkbox"/>
25	to	25	192.168.1.21	25	to	25	TCP ▼	<input checked="" type="checkbox"/>
110	to	110	192.168.1.21	110	to	110	TCP ▼	<input checked="" type="checkbox"/>
587	to	587	192.168.1.21	587	to	587	TCP ▼	<input checked="" type="checkbox"/>
1723	to	1723	192.168.1.21	1723	to	1723	Both ▼	<input checked="" type="checkbox"/>

Figure 15, Ports

Next an application called teamviewer was installed onto the server. This would mean that we can remotely log into the computer from anywhere and make any changes or fix any problems that arose. This meant we could access the server from in college while coding to make any changes needed to the system. Ccleaner and dropbox were also installed. Dropbox was installed to allow us drag and drop any files. FTP was used as the primary method however a few times Dropbox was used as FTP would not include system files. Ccleaner was installed as a tool which helps to keep systems clean and the registry up to date. As a computer does something, it makes a log or creates some temp files. Ccleaner removes any unused files freeing up space and also repairing any bad registry entries. This contributes to keeping a faster and more efficient computer.

After starting into the project, we ran into some problems. In the college, the computers/networks are so locked down that we could not use ftp to connect to the server. Eventually it was decided to set up a VPN connection for the server. This would allow us to

connect to the computer and also access files but also edit them remotely. This proved a more reliable method and looking back should have been used from the start.

For mac, VPN allowed us to mount the XAMPP drive and work remotely off it yet actually only editing the files on the server. This meant that changes to the site were instant. As PHP was our main language, it requires a web server to run the scripts therefore we edited/created php files directly on the server.

The last thing that was set up to finish the system was an update of MySQL/PHPmyadmin and PHP. The versions that were in the latest release of XAMPP were slightly outdated. We opted to upgrade to the latest ones for any added functionality. The server is now running PHP version 5.5.15. After this we then set up the required tables in phpmyadmin and also created the accounts and permissions. Once tables were all set up a backup was taken in case of loss.

It should be noted that one other feature of the system was that a backup of the XAMPP folder was being taken every day just incase the worst was to happen.

6.2 Front End

6.2.1 Article

The article system allows users to view articles by type, date created and also a search system to search through articles. Firstly when the user clicks on articles, the drop down gives the user a list of options such as search, latest and a few categories to choose from. Looking at the search option first, the user is presented with a simple search box and button. When the user enters text into the searchbox, firstly a javascript method will be called which uses jquery's post method send the data to the php script. If the php returns a result, the javascript method will either return the data to a html div. The data will either be the search results or text saying that nothing has matched the search. If the text field was left empty, the method won't even execute the php script and will just output to the user to enter something into the textfield and try again.

```
function SearchArticles(articlesearchdata){
    if (articlesearchdata == ""){
        $('#searchresultdata').html("You must enter something into the search box!");
    }
    else{
        $.post('php/searcharticles.php',[articlesearchdata:articlesearchdata], function(response){$('#searchresultdata').html(response)})
    }
}
```

Figure 16, JavaScript for Search Function

The next options on the list are to view article lists. The first page will show you the latest articles starting with the most recently created article. Very simply, a php script queries the database and returns a list of rows and the data for each row. Then for each row object a view is created with predefined css. In this case, the rows are ordered by id descending.

```
$query = mysql_query("SELECT * FROM `approvedarticles` ORDER BY `articleId` DESC");
```

The next few article pages are creating lists based on the article category eg. Facebook, YouTube and Twitter. More article categories can be added to the site simply as the demand increases. The query for this differs slightly for each page as it is returning a list of rows where the row contains the correct category.

```
$sql = "SELECT * FROM `approvedarticles` WHERE `articleTags` LIKE '%facebook%' OR `articleTitle` LIKE '%facebook%' ORDER BY `articleId` DESC"
```

Here we see that they are ordered again by the newest to oldest and where anything that contains a reference to facebook.

6.2.2 Video

The video system does not differ too much to the article system. The system is designed the exact same way however instead of browsing through articles, its rows of videos. The difference is that in the video system, a video is loaded rather than text/articles. Again the video system has a search function which operates exactly the same.

Handling the data returned in the query is slightly different as we have to load a video file. This is done through storing a link to the folder and then the video file's name. We then are able to create a link out of these two fields. This is done by linking a number of php variables together.

```
if (isset($_GET['videoId'])) {
    $videoId = $_GET['videoId'];
    $query = mysql_query("SELECT * FROM `approvedvideos` WHERE videoId='$videoId'");
    while ($row = mysql_fetch_assoc($query)) {
        $videoTitle = $row['videoTitle'];
        $videoText = $row['videoText'];
        $videoName = $row['videoName'];
        $videoUrl = $row['videoUrl'];
    }
    echo "<h2>".$videoTitle."</h2><br>";
    echo $videoText."<br><br><br>";
    echo "<video controls class='videoplayer' src='".$videoUrl"'></video>";
}
```

Figure 19, Code for Video Search Function

Again the video system consists of the following pages: Search Videos, Latest Videos, Facebook Videos, YouTube Videos and Twitter Videos. In the list, each video link will show an icon, title and small bit of supporting text.

```

while ($row = mysql_fetch_assoc($query)) {
    if ($row != "") {
        $videoId = $row['videoId'];
        $videoTitle = $row['videoTitle'];
        $videoText = $row['videoText'];
        echo "<div id='videolist' class='col-md-12'>";
        echo "<div id='videoicon' class='col-md-3'>";
        echo "<a href='watchvideo.php?videoId=$videoId'><img src='include/images/thumbnail.png' width='100%'></a>";
        echo "</div>";
        echo "<div id='videotext' class='col-md-offset-1 col-md-7'>";
        echo "<div class='col-md-8><div class='caption'><h4><a href='watchvideo.php?videoId=$videoId'>$videoTitle</a></h4>";
        echo substr($videoText, 0,200).".....</div><br><br><br>";
        echo "</div>";
        echo "</div>";
    }
}

```

Figure 18, Code to Display Videos

6.2.3 Forum

The forum consists of two main pages for the user. After clicking on the forum you will be brought to the search/display page. Here you can search for topics using keywords but also you will be presented with a list of forum topics ordering in newest to oldest.

The search uses the same architecture as discussed before in the article section. The search is looking for keywords in titles and any supporting text. After a list of topics are displayed for the user to click on.

How can I check to see if my kid signed up for something online?

Date Created: 27-02-15

Any help would be appreciated!

Comments

The Internet is full of enticements to sign up for something. Sometimes these come-ons make it sound as if you've won a prize and all you need to do to receive your winnings is register. If your kid registered for an account, downloaded a program, or entered a contest, you may not know unless you notice a new program on your computer. Here's how to get to the bottom of things, starting with the least intrusive: Ask your kid. Explain that there are a lot of scams out there that can spread viruses, so it's important that they ask permission before signing up for something. Check the browser history. Look for telltale words such as "registration page" or "thank you for registering." Check your kid's email. Companies sometimes send a confirmation email to the address used in the sign-up. If your kid is under 13, kid-targeted websites and apps are supposed to get your permission before they collect or use any personal information from your kid (although it's easy for kids to dodge any age restrictions and say they're older). Hope this helped!



Comment **Return**

Figure 19, Topic Page

After entering a topic the user is able to view the topic but also place comments underneath. This may be to help other users find a solution to the question/topic they have asked about.

The user also has the option to create new topics through the simple form. This form will automatically add to the database.

6.2.4 Contact

The contact page is very basic. It allows a registered/non registered user a method of getting in contact with the admin staff. It simply adds a row into a database table which is viewable as a list in the admins menu. An admin can then review the message and delete after the message has been answered.

6.2.5 Content Creator Form

Firstly as a content creator you get access to another section of the website. Once your account has been updated to give you your new user privileges you will get an extra menu option. Here you have a link to two forms. One form is to create a new article and the other is to add a new video to the site.

On selecting to add a new article, you are brought to that form. There you must assign a new Title, Body and Article Tags. For the article body, we used a text editor called TinyMCE. This javascript plugin allows plain text to be converted into html code. This means that linebreaks and <p> tags are automatically put in. This is crucial to get the article to display properly on the page without manual formatting.

After this some tags must be put into the article. This is primarily for searching and also categorizing the articles.

The second form is less detailed. The following fields must be filled out: Video Title, A small video description, a file which must be of .mp4, select a group tag and also put some free-text tags.

Video Title

Video Description

Select File
 No file chosen

Select Group Tag

Video Tags
 Use spaces to separate tags

Figure 20, Add video form

6.2.7 Admin Functionality

The admin menu has five different choices:

Article Approval List

Here the admin can view all the article submissions from content creators. Each article must be approved before it becomes public on the site. On this page, there is a list ordered by oldest to newest meaning the oldest articles that are waiting for approval appear first.

When an admin clicks on an article to view, all the input fields are displayed followed by an accept and reject button. If the admin accepts the article, the row is copied to a new table which stores approved articles. If the admin rejects it, it is just deleted. Unfortunately time was not on our side to create the notification system. This would have let users know if the article was accepted/rejected.

Video Approval List

Again the video approval list works the exact same way that the article approval list works. The difference is that the admin can watch the video but the remainder of the method is identical.

Administrate Forum

The administrate forum enters you into a different version of the forum. Beside each article or comment there is a delete button. This is to remove any unwanted posts/bad behaviour or explicit content created that does not agree with our policy and standards. Pressing the delete button will either remove the comment or the article and all comments associated with the article.

View User List

The user list allows an admin to remove a user or change the users privileges. This allows an admin to change a regular user to a content creator.

View Contact Us Submissions

This page shows all the submissions through the contact form. An administrator can view the list and after the appropriate action has been taken, they can delete the entry. This list is ordered by id ascending meaning that the oldest ones will appear at the top of the list as they will be a higher priority due to the first come first serve basis.

6.3 Development Environment

6.3.1 PHP

PHP is a server side language. For our site, php did all the talking between the front end and the database.

6.3.2 SQL

SQL is used to hold all the sites data. In this case it stores articles, forum topics, some but not all the data for videos and also the users.

6.3.3 HTML

HTML was used mainly to construct the layout of the site.

6.3.4 Bootstrap

Bootstrap was used to create all of the design/layout of the site alongside HTML. Bootstrap is an open source CSS framework designed for fast prototyping.

7. Testing

In this section we will discuss how we went about testing and fixing any bugs that popped up in the system either during development or after development during user testing.

7.1 Unit Testing

The system is broken up into many subsystems. They are all created before implementing them into the full website with design. Each systems functionality was tested before placed into the main system. This speeded up the process of implementing the sub-system into the main website.

After each system was dropped into the main site, the system was re-tested to look for new bugs. As to be expected we encountered many bugs. Most of these were just small conflict with other bits of code from other systems as when they were all operating in the one site, some methods overlapped.

One major bug that was encountered was trying to stop users accessing pages based on their account privileges. This was quickly rectified however with a different if statement

```
if ($role !== "Admin" && $role !== "Content") {  
    header("Location: noaccess.php");  
}
```

The rest of the site came together easily enough. Simple things like commas in the wrong place were the main source of bugs.

The other main bug that was encountered was after the completion of the site. Instead of copying a row from one table to another, the script was taking everything out of the row into the script and then re-submitting it into a new row. This was causing problems and was solved by using an sql query to copy the row and not bring any data to the script.

7.2 Usability Testing

To test if the site was fit from a usability point of view, user testing was carried out in the form of a questionnaire. Neighbours with children were sat in front of a laptop with the site's home screen open. From there they were given a set of instructions to follow along with sections they could input their own opinion of the site.

Users were asked to:

- Register an Account
- Login with the created account
- Create a forum topic
- Search for an existing forum topic

- Find an existing article
- Find an existing video
- Send a message on the contact form
- logout

The users account was then granted content creator privileges and were asked to create an article and submit a sample video for the admins approval.

After reviewing the questionnaires only a few minor issues were addressed, mainly to do with the navigation of the sites functions. Some users had issues with the topics menu as the 'Search' and 'Create' button were not clearly labelled.

When asked about the sites design, mixed opinions were received. Some users found the sites design simple and clean while others found it bland and boring to look at. To solve this issue more colour was added to the site as well as a more interactive main navigation bar, as seen in figure 21.

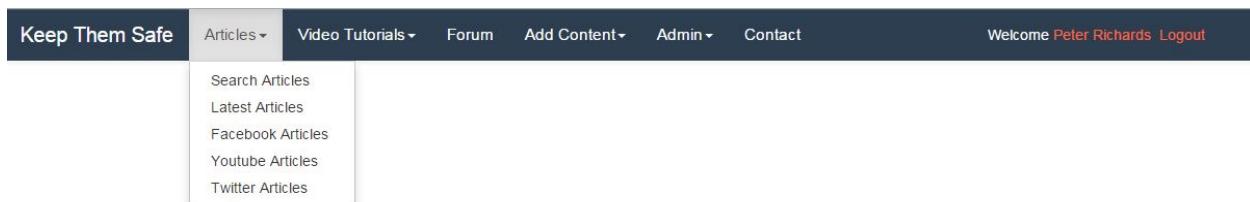


Figure 21, Main Navigation

7.3 Test Plan

The test plan is rather simple due to the design of the system. As the system is broken into sections that should at the end fit together, as each sub system is built it was tested for problems. After each system was tested and confirmed fully functional, it was merged into the main site architecture. It needed to be tested for integration and any possible problems that may arise with merging the systems into the main site.

An example of how a test plan worked as follows:

The comment system is similar to a forum. A user may ask a question or request help and users are able to respond to the post in a forum style situation. The way that it's tested is to try and complete the two different parts. The first part is to be able to create a new topic. The user should then be able to refresh the page and see the topic in the list of topics. The second part of the sub system is to be able to comment in the topic. The user will be able to click on the topic to view it in more detail as they are just presented with a title/question on the list view. In the detailed view they will be able to add a comment to the topic. After following the form, if successful the comment will be added to the database and the user will be able to see it on the page.

Article System is another sub system that was created. Here a content creator is able to create a new article through an html form. This is text based content and is stored in a database table. The testing method for this is as follows: If successful, the user should fill out the form and see the new article populated on the list view.

Testing the system integration was the most difficult task. As the entire subsystems are completed, they can be added to the main system one at a time. If all the subsystems are added at once, it will be harder to debug the system as finding errors can take much longer. If each system was to be implemented to main system one at a time, fixing errors would become a much easier task as there is less code to debug.

The plan for testing the system is to test each sub-system first and once each system is working to implement it into the main system. This will save time in the long run and reduce the time spent working out issues.

User testing was also done in the form of a questionnaire with the aim to see if our sites usability was good and if anything could be improved on it.

The questionnaire can be found in the appendices at appendix 10.

7.4 Result

At the end of the build, the system overall did not provide too many bugs. The majority of the bugs encountered were simple to fix as it was usually only caused by syntax errors like having a misplaced comma or not linking the site to the correct database table correctly. These bugs could have been much more difficult to find if we had not used unit testing, as spotting errors was much easier due to the smaller amount of code per unit. Through user testing we were able to determine flaws with the site that we may not have realised on our own as we knew its layout inside out. The questionnaire also helped to make the site less bland for users allowing us to add colour and responsive attributes.

8. Conclusion

8.1 Peter Richards

For me, the project was a great eye opener. Being able to create something from start to finish and put all the skills that we have learnt from year one to the test was a great reward. The project really helps you understand what you have learnt and for me has also guided me towards the two main areas of computer programming which would be PHP and Database Management.

After completing the project, you automatically know ways you could improve it and also the different ways that you would go about creating it and ultimately saving time. Many new methods and also systems that could have been implemented now come to mind such as

- Private Message System
- Video Rating System
- View Count System

Looking back, these systems would have been very simple to implement. These systems would have used the same architecture as the comment system or a similar method.

Testing the system ended up being much easier than I originally thought due to the option we took of checking each subsystem before integrating into the main site. During the integration however we did add extra things to the subsystems as we figured we could get more out of what we had originally created.

It was good to work in pairs as it gave us an insight into what working in a group would be like as this was the first big project that we had taken on. The project definitely gave me an insight into the type of role that I would like to pursue in computer programming later down the track.

It would have been nice to have more time to work on code, as we got to the end of the project we started coming up with all these little extra ideas that we could have implemented if we had more time. This definitely taught me more about sticking to deadlines as on a few occasions we probably got a little too relaxed or other assignments got in the way of the project. Better time management and project planning in the future would fix this.

8.2 Jonathan Dempsey

In the early stages of the project we had a pretty good idea of what our main aim would be and had knew that we would need to use PHP and a database. I remember feeling a bit apprehensive about the project at first because although I had worked with PHP and databases before, my experience with them was minimal and I was mediocre at best with them. As a whole the project was overwhelming for my skill level, but as time went on and with Peter's help I realised that the project was going to be very manageable.

We started from square one and began implementing the sites functions separate from one another, starting simple and slowly adding more complicated parts and made sure they worked on their own. I feel this was a good way to go about the implementation as it really allowed us to fully understand how everything worked before it got too messy to understand when it was all to be put together.

I found working with Peter to be a good experience and thought we worked well together. If one of us got stuck with something the other was usually able to successfully help solve the problem. I also feel working in pairs was a good way to see what it would be like to work on a project with others out in the real world and highly value the experience.

In the end we achieved the majority of our goals for the project. Though had we more time I would've liked to fully flesh out a better design for the website and added a few smaller functionalities that would not have been necessary to add, but would have been nice additions.

8.3 Discussion

This section will outline a short discussion about the project. The discussion will include what we created, the steps we took to create the site, what we had to research in order to carry out the project, the results and what we could have added given more time.

The projects goal was to create a website that allows parents/guardians to learn about web safety for their children. We feel the project achieved its goal as the site allows users to create an account and view articles and videos about Internet safety as well as ask/answer questions others have about Internet safety in the topic forum. Given the correct privileges, a user can create articles and submit videos about Internet safety to an approval list for validation by the sites administrators. Should a user have a question they want to ask the admins directly, there is a contact form they can fill out with any questions or problems. Finally, a few users are given full privileges, which allows them to approve or disapprove pending videos and articles into the main site. They also have the ability to give or take privileges away from users as well as delete negative comments or topics from the forums.

The majority of the sites functionality comes from the use of PHP interacting with the web server. We both had a small amount of experience with PHP from the previous year of college

which included, adding/updating/deleting data from a server, was enough to have a decent understanding of how PHP worked, but much more research had to be done to achieve the projects goal. Both of us were provided with an avaya server but decided against using that for our project, instead we used Peter's web server and worked off that for the entirety of the project. The first stages of the project mainly consisted of learning and researching PHP, SQL and how they would work with the web server.

After becoming familiar with all the technologies and languages, we began to implement the sites functionality. Each of the sites functionality was built separately from all the other functions and was only added to the main site once we were sure it worked flawlessly. Through this unit testing, we ran into very little problems in terms of the projects progression (Most of which is detailed in 7.3).

The result was more or less what we had anticipated, we implemented all the major functionality we had envisioned from the beginning of the project. It is now only a matter of populating the site with articles and videos for users to view and learn from, which will be done after the project is submitted. The sites design isn't quite what we expected as most of the focus was on coding and not as much focus on the design.

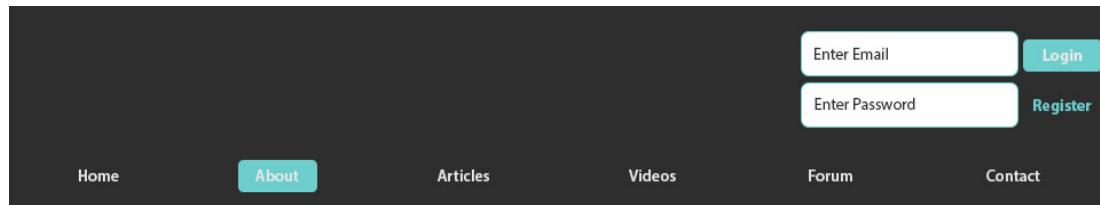
There was additional functionality we would have liked to implement, though not necessary, would have given the website a bit more life. Some of these functions were mentioned by user testing such as having a view count, a rating system for the content, the ability to search data based on view count and most popular and private messaging. We could have also put a bit more effort into the sites design, while not bad, could have used a bit more attention and editing.

In conclusion, We are both pleased with the way the project turned out and are glad to have had the experience of creating it.

9. References

1. [Online] <http://www.liv.ac.uk/~maryrees/homepagemath302/PlagiarismStatement.pdf>
2. [Online] <http://www.tinymce.com/>
3. [Online] http://www.phpmyadmin.net/home_page/index.php
4. [Online] <http://www2.amk.fi/digma.fi/www.amk.fi/material/attachments/vanhaamk/etutanto/0303011/5gxhOAL0I/HierarchicalDatabaseModel.jpg>
5. [Online] http://www.comphist.org/computing_history/new_page_9.htm
6. [Online] http://www.fing.edu.uy/inco/grupos/csi/esp/Cursos/cursos_act/2000/DAP_DisAvDB/documentacion/OO/Evol_DataModels.html (Asbjorn danielsen)
7. [Online] <http://4840895.blogspot.ie/2009/04/history-of-dbsms.html> (Carl Chebli, 2009)
8. [Online] <http://wp.sigmod.org/?p=688> (Thomas Haigh, 2012)
9. [Online] <http://img.c4learn.com/2012/02/Relational-Database-Management-System.jpg>
10. [Online] http://www-03.ibm.com/ibm/history/exhibits/builders/builders_codd.html (IBM Research News, 2003)
11. [Online] <http://databases.about.com/od/sql/a/What-Is-Sql.htm>
12. [Online] <http://php.net/manual/en/history.php.php>
13. [Online] http://www.nusphere.com/php/php_history.htm
14. [Online] http://www.phpmyadmin.net/home_page/about.php
15. [Online] <http://getbootstrap.com/about/>
16. [Online] <http://www.blooberry.com/indexdot/history/html.htm>
17. [Online] <http://www.yourhtmlsource.com/starthere/historyofhtml.html>

10. Appendices



“Internet safety begins at home”

A site dedicated to help you and your children be safer on the web

Newest Articles

Article One	...
Article Two	...
Article Three	...
Article Four	...
Article Five	...
Article Six	...
Article Seven	...
Article Eight	...

Newest Videos

Video One	...
Video Two	...
Video Three	...
Video Four	...
Video Five	...
Video Six	...
Video Seven	...
Video Eight	...

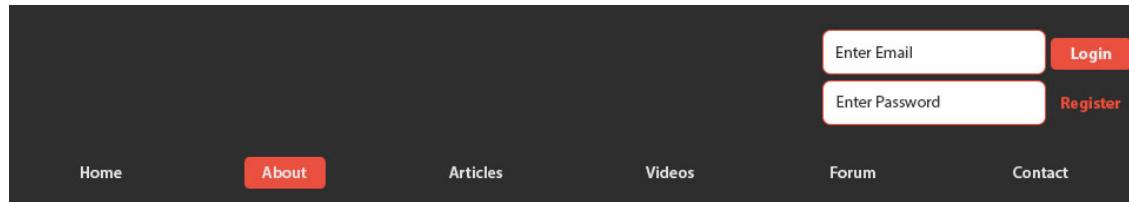
Social Media



Tweets

The screenshot shows a social media feed with two tweets from the account @TheOnion. The first tweet is from 'The Onion' (@TheOnion) and reads: 'From The Archives: Government Official Who Makes Perfectly Valid, Well-Reasoned Point Against Israel Forced To Resign' with a link to onion.com/1BUU2AC. The second tweet is also from 'The Onion' (@TheOnion) and reads: 'I'm just glad that my loved ones now have a plan in place to rack up enormous legal costs' with a link to onion.com/1zbwzr and a picture of Donald Reubens. Below the tweets, there is a 'COMMENTARY' section featuring a photo of Donald Reubens and the text: 'If I End Up On Life Support, My Family Knows The Type Of Long, Protracted Legal Battle I Would Want' followed by 'BY DONALD REUBENS'. At the bottom, there is an 'Expand' button and another tweet from 'The Onion' (@TheOnion) about a Mexican restaurant giving a twist to Mexican cuisine.

[1] Figure 22, Wireframe 1



“Internet safety begins at home”

A site dedicated to help you and your children be safer on the web

Newest Articles

Article One	...
<u>Article Two</u>	...
Article Three	...
Article Four	...
Article Five	...
Article Six	...
Article Seven	...
Article Eight	...

Newest Videos

Video One	...
Video Two	...
Video Three	...
Video Four	...
Video Five	...
Video Six	...
Video Seven	...
Video Eight	...

Social Media

Tweets

The Onion @TheOnion From The Archives: Government Official Who Makes Perfectly Valid, Well-Reasoned Point Against Israel Forced To Resign onion.com/1BU12AC
12m

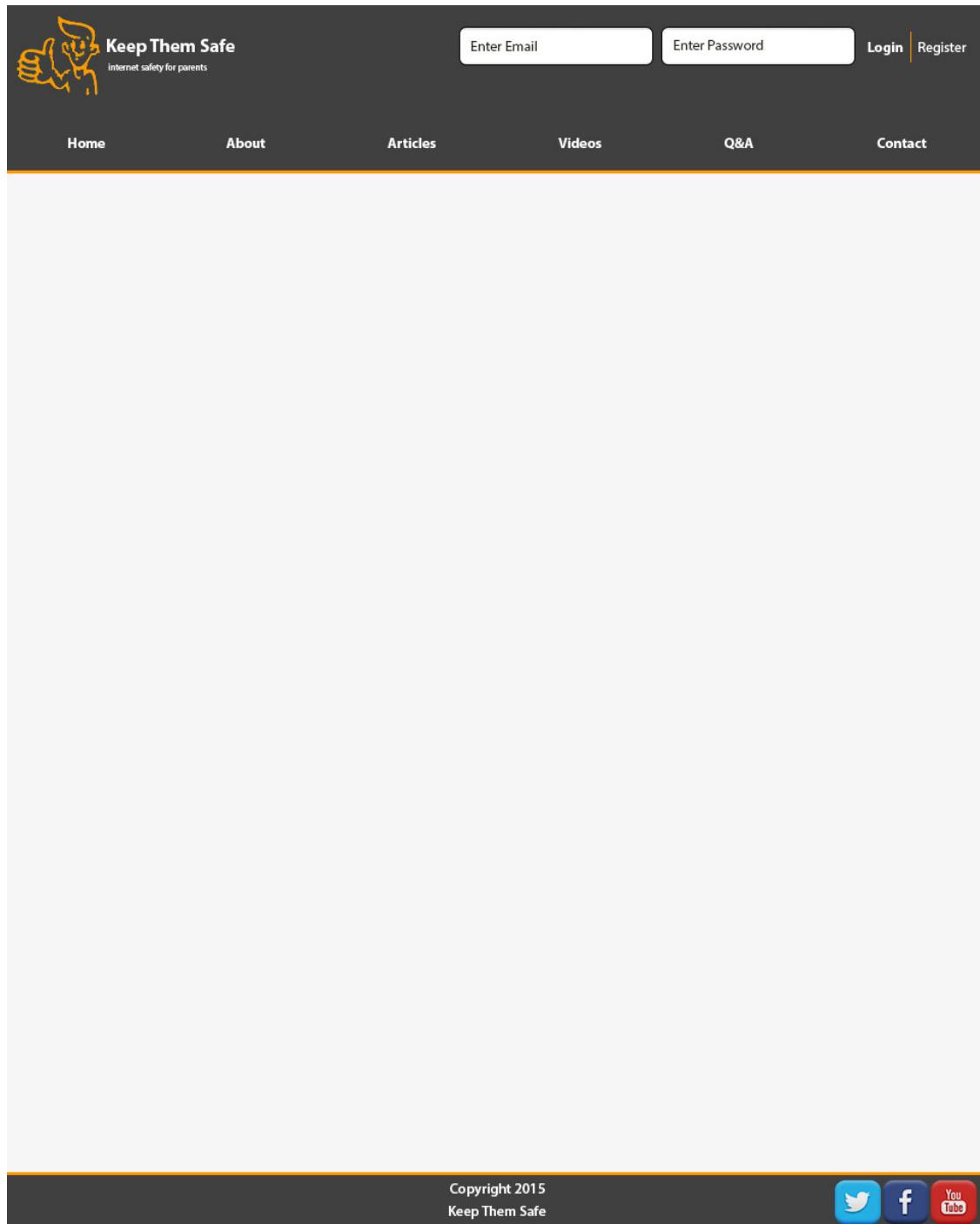
The Onion @TheOnion "I'm just glad that my loved ones now have a plan in place to rack up enormous legal costs." onion.com/1zbwzr pic.twitter.com/76aFkMwQhV
45m

COMMENTARY

If I End Up On Life Support, My Family Knows The Type Of Long, Protracted Legal Battle I Would Want

BY DONALD REUBENS

[2] Figure 23, Wireframe 2



[3] Figure 24, Wireframe 3



Copyright 2015
Keep Them Safe



[4] Figure 25, Wireframe 4

Grumpy wizards make toxic brew for the evil Queen and Jack.

One morning, when Gregor Samsa woke from troubled dreams, he found himself transformed in his bed into a horrible vermin. He lay on his armour-like back, and if he lifted his head a little he could see his brown belly, slightly domed and divided by arches into stiff sections.

"The bedding was hardly able to cover it and seemed ready to slide off any moment. His many legs, pitifully thin compared with the size of the rest of him, waved about helplessly as he looked."

[5] Figure 26, Abel Font

Grumpy wizards make toxic brew for the evil Queen and Jack.

One morning, when Gregor Samsa woke from troubled dreams, he found himself transformed in his bed into a horrible vermin. He lay on his armour-like back, and if he lifted his head a little he could see his brown belly, slightly domed and divided by arches into stiff sections.

"The bedding was hardly able to cover it and seemed ready to slide off any moment. His many legs, pitifully thin compared with the size of the rest of him, waved about helplessly as he looked."

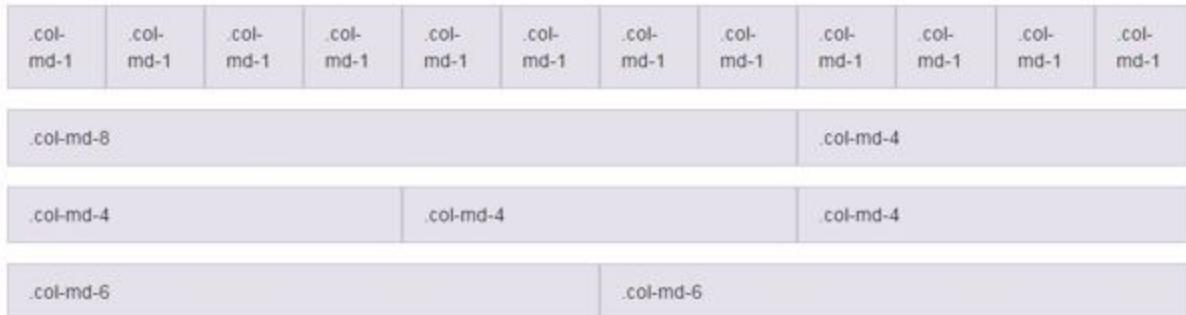
[6] Figure 27, Lato Font

Grumpy wizards make toxic brew for the evil Queen and Jack.

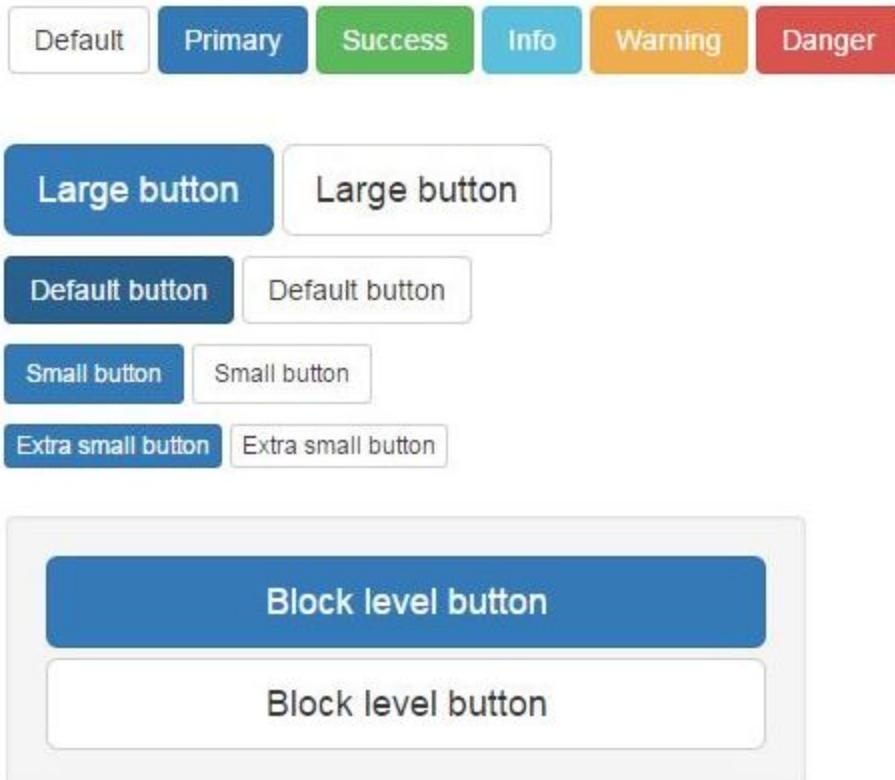
One morning, when Gregor Samsa woke from troubled dreams, he found himself transformed in his bed into a horrible vermin. He lay on his armour-like back, and if he lifted his head a little he could see his brown belly, slightly domed and divided by arches into stiff sections.

"The bedding was hardly able to cover it and seemed ready to slide off any moment. His many legs, pitifully thin compared with the size of the rest of him, waved about helplessly as he looked."

[7] *Figure 28, Questrial Font*



[8] *Figure 29, Bootstrap Columns*



[9] Figure 30, Bootstrap Buttons

- 1. Register an account**

- 2. Log into the account you just created**

- 1. I found registering an account was clear and easy to follow**

Strongly Disagree	Somewhat Disagree	Neither	Somewhat Agree	Strongly Agree

- 2. The register form could use some improvements**

Strongly Disagree	Somewhat Disagree	Neither	Somewhat Agree	Strongly Agree

- 3. If you agree with 2, which improvements could be implemented?**

-
4. **I was able to log in without any problems**

Strongly Disagree	Somewhat Disagree	Neither	Somewhat Agree	Strongly Agree

5. **The login function could use some improvements**

Strongly Disagree	Somewhat Disagree	Neither	Somewhat Agree	Strongly Agree

6. **If you agree with 5, which improvements could be implemented?**
-

1. **Create a forum topic of your choosing**

2. **Search for a forum question ‘Locking down my sons computer?’ and post a comment**

1. **I created a topic on the forum without any issue**

Strongly Disagree	Somewhat Disagree	Neither	Somewhat Agree	Strongly Agree

2. **Adding a topic could use some improvements**

Strongly Disagree	Somewhat Disagree	Neither	Somewhat Agree	Strongly Agree

3. **If you agree with 2, which improvements could be implemented?**
-

4. **I found the topic ‘...’ and commented on the topic easily**

Strongly Disagree	Somewhat Disagree	Neither	Somewhat Agree	Strongly Agree

5. The adding and commenting on a topic could use some improvements

Strongly Disagree	Somewhat Disagree	Neither	Somewhat Agree	Strongly Agree

6. If you agree with 5, which improvements could be implemented?

1. Find an article called 'Facebook – The Basics'

2. Find a video called 'Searching on Youtube'

1. I found the article '...' without any issue

Strongly Disagree	Somewhat Disagree	Neither	Somewhat Agree	Strongly Agree

2. Searching for an article could use some improvements

Strongly Disagree	Somewhat Disagree	Neither	Somewhat Agree	Strongly Agree

3. If you agree with 2, which improvements could be implemented?

4. I found the video '...' without any issue

Strongly Disagree	Somewhat Disagree	Neither	Somewhat Agree	Strongly Agree

5. Searching for a video could use some improvements

Strongly Disagree	Somewhat Disagree	Neither	Somewhat Agree	Strongly Agree

6. If you agree with 5, which improvements could be implemented?

1. Sent a message through the contact form

2. Logout of your account

1. I used the contact form without any issue

Strongly Disagree	Somewhat Disagree	Neither	Somewhat Agree	Strongly Agree

2. The contact form could use some improvements

Strongly Disagree	Somewhat Disagree	Neither	Somewhat Agree	Strongly Agree

3. If you agree with 2, which improvements could be implemented?

4. I logged out without any issue

Strongly Disagree	Somewhat Disagree	Neither	Somewhat Agree	Strongly Agree

5. Logging out could use some improvements

Strongly Disagree	Somewhat Disagree	Neither	Somewhat Agree	Strongly Agree

--	--	--	--	--

6. If you agree with 5, which improvements could be implemented?

1. Create an article using the content creator account provided

2. Add a the sample video using the content creator account provided

1. I created an article without any issue

Strongly Disagree	Somewhat Disagree	Neither	Somewhat Agree	Strongly Agree

2. The contact form could use some improvements

Strongly Disagree	Somewhat Disagree	Neither	Somewhat Agree	Strongly Agree

3. If you agree with 2, which improvements could be implemented?

4. I added a video without any issue

Strongly Disagree	Somewhat Disagree	Neither	Somewhat Agree	Strongly Agree

5. Adding a video could use some improvements

Strongly Disagree	Somewhat Disagree	Neither	Somewhat Agree	Strongly Agree

6. If you agree with 5, which improvements could be implemented?

[10] Figure 31, User Questionnaire