Halil Olsen (16849180)

Project supervisor: Jarod Locke | Second reader: Goran Soldar

Interactive and Educational Web Experience on Substance Abuse

# Acknowledgments

Throughout the development of this project, I have received a great deal of support and assistance from several people. Firstly, I would like to thank my supervisor, Jarod Locke, whose expertise and continued support was invaluable.

I would also like to acknowledge my mother, brother, grandfather, and girlfriend. Without their love and support, I could have never returned from my intermission in 2019 and gone on to finish my third year of university. I am, and always will be, grateful for them.

# Abstract

This project was created with the intention of educating users on the statistics, effects, and realities of substance abuse in the United Kingdom. I aimed to develop the front and back-end of a website, using HTML, CSS, PHP, and JavaScript. I wanted the website to be dynamically updateable through the back end to update the information on the website if, and when, it changes.

I gained a lot of valuable experience in all areas of development for this project. I only had a small amount of experience in web development, and all that encompasses it, at the beginning of this project. This project threw me in the “deep end” of web development and as such forced me to quickly, and efficiently, learn the necessary techniques and skills required to implement the vision I had for the website.  
I learnt a lot about managing a project efficiently during this time, as well as managing the risks and problems that presented themselves along the way.

The outcome of this project that has impacted me personally the most was discovering the number of possible problems that can arise when trying to design and implement a website with this kind of functionality.   
I went into this project with a fair amount of experience in HTML and CSS, mitigating the simpler aspects of the development of the website.   
Learning PHP while developing in it did prove to slow the production of the website down at times and ultimately did not allow me to implement the full functionality I was hoping to achieve in the time I was given. The same thing could be said for JavaScript, however it was used far less in my project than PHP.

The end product is a comprehensive representation of my progress in these areas and, despite the room for potential further development and improvement, is something that I would regard as a completed version one release.

**Link to website:** <http://hbo13.brighton.domains/hbo13_website/>

**Link to presentation video:**

Contents

[Acknowledgments 1](#_Toc42129213)

[Abstract 1](#_Toc42129214)

[Introduction 4](#_Toc42129215)

[Methodology 5](#_Toc42129216)

[Project Management 5](#_Toc42129217)

[Agile 5](#_Toc42129218)

[Alternatives 5](#_Toc42129219)

[Primary Research 6](#_Toc42129220)

[Secondary Research 7](#_Toc42129221)

[Secondary Research For Website Content 7](#_Toc42129222)

[Secondary Research During Implementation 8](#_Toc42129223)

[Development Tools 9](#_Toc42129224)

[IDEs 9](#_Toc42129225)

[cPanel 9](#_Toc42129226)

[phpMyAdmin 10](#_Toc42129227)

[MySQL Databases 10](#_Toc42129228)

[Testing 11](#_Toc42129229)

[Methodology 11](#_Toc42129230)

[Test Plan 11](#_Toc42129231)

[Product Description 11](#_Toc42129232)

[Introduction 11](#_Toc42129233)

[Requirements Analysis 12](#_Toc42129234)

[Design 13](#_Toc42129235)

[User Interface Design 13](#_Toc42129236)

[Three-Tier Architecture 14](#_Toc42129237)

[Database Design 14](#_Toc42129238)

[Implementation 15](#_Toc42129239)

[Modals 15](#_Toc42129240)

[The Quiz 18](#_Toc42129241)

[Evaluation 20](#_Toc42129242)

[Critical Review 20](#_Toc42129243)

[Risk Analysis 20](#_Toc42129244)

[Acute risks 20](#_Toc42129245)

[Chronic risks 21](#_Toc42129246)

[COVID-19 pandemic 21](#_Toc42129247)

[What has been successful? 22](#_Toc42129248)

[What could I improve upon? 22](#_Toc42129249)

[The future 22](#_Toc42129250)

[References 23](#_Toc42129251)

[Appendices 24](#_Toc42129252)

[Appendix 1 – Record of Meetings With Supervisor 24](#_Toc42129253)

[*27th September* 24](#_Toc42129254)

[*4th February* 26](#_Toc42129255)

[*3rd April* 27](#_Toc42129256)

[*5th May* 27](#_Toc42129257)

[*11th May* 27](#_Toc42129258)

[*19th May* 27](#_Toc42129259)

[Appendix 2.1 – Test Plan Template 28](#_Toc42129260)

[Appendix 2.2 – Test Plan Example 1 28](#_Toc42129261)

[Appendix 2.2 – Test Plan Example 2 29](#_Toc42129262)

[Appendix 2.3 – Test Plan Example 3 29](#_Toc42129263)

[Appendix 3.1 – Homepage UI Wireframe Option 1 30](#_Toc42129264)

[Appendix 3.2 – Homepage UI Wireframe Option 2 30](#_Toc42129265)

[Appendix 3.3 – Subpage UI Wireframe Option 1 31](#_Toc42129266)

[Appendix 3.4 – Subpage UI Wireframe Option 2 31](#_Toc42129267)

[Appendix 3.5 – Quiz UI Wireframe Option 1 32](#_Toc42129268)

[Appendix 3.6 – Quiz UI Wireframe Option 2 32](#_Toc42129269)

[Appendix 3.7 – Quiz Wireframe Option 3 33](#_Toc42129270)

[Appendix 4.1 – Example Modal Code from W3Schools 33](#_Toc42129271)

[Appendix 4.2 – Modal Class CSS Code 34](#_Toc42129272)

[Appendix 4.3 – Modal-Content Class CSS Code 34](#_Toc42129273)

[Appendix 4.4 – JavaScript For Modals 34](#_Toc42129274)

[Appendix 4.5 – Example of Quiz Question In Modal 36](#_Toc42129275)

[Appendix 5.1 – Database Connection PHP Code 36](#_Toc42129276)

[Appendix 5.2 – Example of Displaying A Question and Its Answers through PHP 36](#_Toc42129277)

[The SQL queries 36](#_Toc42129278)

[Displaying the question as a heading 37](#_Toc42129279)

[Displaying the answers as labels for radio buttons 37](#_Toc42129280)

[Appendix 5.3 – The handling of the user’s answers in marijuana\_dest.php 38](#_Toc42129281)

[Appendix 5.4 – 0 answers answered correctly in a quiz code example 38](#_Toc42129282)

[Appendix 6.1 – Questionnaire Template 40](#_Toc42129283)

[Appendix 6.2 – Questionnaire results 46](#_Toc42129284)

# Introduction

I started this project with a vision of a fluid, interactive and educational web experience that details and explores different areas of substance abuse in this country. I wanted it to be easily updateable via a back-end system as statistics on this topic can, and do, change often and significantly.  
I felt that there was a gap in the “market” for a website like this. Looking online for advice, statistics or more general information will more than likely lead you to a government or organisation’s website. These have a cold ambience and wording more akin to a scientific report than a conversation about substance abuse.

The main goal of this project therefore was to fill this gap and provide people with a website where they can find the most up to date information and advice in a warm and aesthetically pleasing environment. I wanted my targeted demographic, 18 to 24-year olds, to have the option of a website that is not solely focused on just the statistics of substance abuse or poses as a source of advice that just provides links that help you find a local GP or centre.

Another important goal of the project that I would not consider secondary to the above is the ability for it to be updated quickly and easily by anybody with the administrator credentials. Information on this topic changes quickly and sometimes significantly, so it is important that the website be kept up to date and the task of doing so be simple and quick. It may not be the case that the person updating the website has any experience in front or back-end web development at all, so making it as simple as possible for them to alter the information on the website is hugely important.

Smaller goals of the project include learning a number of techniques that are completely new to me as a developer and Computer Science student. Before I started this project, I had never developed in PHP or JavaScript and had never used MySQL Databases and PHPMyAdmin.   
Learning these languages and techniques in order to fulfil the rest of my goals was essential and would allow me to broaden my skills in a few different areas of software development.

# Methodology

## Project Management

### Agile

In my interim planning and investigation report, I explained that I chose to approach this project using an agile project management method as well as outlined some of the reasons why I chose to use it. Below, I will discuss this approach and explain my reasons further and discuss the effects that it had on my project as a whole.

Agile project management is an iterative approach to delivering a project, life cycles are composed of the release of several iterations or sprints which work towards the completion of the project (Association for Project Management, 2020). This approach allowed me to develop, test and, if needed, adjust each sprint release.  
In my interim planning and investigation report, I mentioned that most of the sprint releases will represent some functionality on the website and this turned out to be true. One thing that did change after I wrote the report were the sprints themselves, there were a lot of sprints that were needed that I did not initially foresee as well as issues that arose during these sprints. To cover this, I provided an updated schedule of activities in my appendix.

Using this approach undoubtably helped me deal with all the issues that I encountered throughout the development of this project. Having the ability to prioritise certain tasks over others and leave issues to be fixed in the future while another task is completed helped to keep my motivation and productivity up.

I tried to make sure that my development was test-driven, by which I mean that I would test every component and feature during and after its development to validate its functionality. Doing this before moving onto the development of another component helped to minimise arising issues and, even if they did, they were often much less significant than they could have been due to this test-driven approach.  
I also tried to make sure that I was developing in small sprints. This decreased the size of each individual component I would be developing and, therefore, decreased the time that I would have to spend testing each component. Sometimes this was not an option available to me, for example writing the SQL queries that would enter the questions and answers for the quizzes is a time-consuming task. Luckily, it is easily tested by running some simple GET SQL queries.

### Alternatives

There are many popular project management methods, each with their advantages and disadvantages and each suited more to different, specific project needs.   
For instance, the Six Sigma methodology aims to improve quality by reducing the number of errors in a certain process by identifying what is not working and removing it. It uses quality management methods which are mostly empirical and statistical (Zenkit, 2018).  
This methodology is much more suited to projects conducted by large companies and organisations that want to improve the quality and efficiency of their processes in a data-driven way. Therefore, it was not a methodology I considered for this project.

An alternative I did consider, however, is the Waterfall methodology, one of the more traditional methodologies. It is a linear, sequential design approach where progress flows downwards, in one direction. If I were to employ this methodology, I would only be able to move onto the next phase of development once the current phase had been completed, and only in the specific order below:

* System and software requirements
* Analysis
* Design
* Coding
* Testing
* Operations

There is little flexibility in this methodology. I could not, for instance, change my decision on the design of the website after undergoing testing. It also limits the amount of possible customer feedback available to the developer during the development process. In my case, this would not be so much of an issue as I am a solo developer and there is no one specific customer for this project. The lack of flexibility, however, was the key factor that turned me away from this methodology.   
Waterfall is better suited to larger projects that require maintaining stringent stages and deadlines or projects that have been conducted many times where chances of problems arising during development are low.

## Primary Research

I did not require a lot of primary research to be undertaken for my project as all the content that would be featured on the website would be the result of my secondary research. I did however think that it was important to conduct primary research to assist me in the user interface decision on the website.   
I decided to create a simple questionnaire for this and send it out to my fellow Year Three Computer Science students via email, there a number of advantages to a questionnaire which I have outlined below:

* They are quick and free to send out
* Large populations can be targeted easily
* Respondents can reply easily

There are, however, also several disadvantages that come with email questionnaires:

* They may be filtered as spam and easily deleted
* If the questionnaire is sent as an attachment, it may not be readable or take too long to download
* If the questionnaire is embedded in the email, it may not appear clearly as formats change from one system to another
* It can be difficult for respondents to reply anonymously, which may put them off

(Dawson, 2015)

These advantages were the main reason I chose to send the questionnaire out via email, it was quick and straightforward for me to do so allowing me to focus more time on other parts of the project but I could also easily send it to every Computer Science student.  
I did encounter some of these disadvantages however. I was worried that I would not get a lot of respondents as I understood that everyone was busy with their own projects. I was hoping that, as we were all in the same position, a good number of people would reply with their answers, however. I was also worried that many people would not reply with their answers as it could not possibly be anonymous due to them having to use their university email. The obvious way to fix this problem would be to use something like SurveyMonkey so that all the answers could be anonymous. However, as SurveyMonkey is not GDPR compliant, I could not use it.

These worries came to fruition in the end as I only received two questionnaires back from my email. This, of course, is not enough for me to draw any conclusions from. As the questionnaire was only asking for UI input and did not require any Computer Science knowledge, I decided to ask some family members for their answers. I ended up with ten replies in total and, given the circumstances, I was happy with this outcome.  
My results from the questionnaire, as well as the questionnaire template itself, are shown in the appendix.

Another form of primary research that I conducted was using Stack Overflow. I was able to use Stack Overflow whenever I faced challenges throughout the implementation. This allowed to receive information, or answers, that are specific to my problem and context. The ability to receive the help and feedback from people who are much more experienced than I am in the area that my question refers to (PHP, MySQL, JavaScript, etc.) was invaluable.

One example is when I was facing a problem implementing my quizzes within modals. I wanted to use JavaScript store the user’s score in a variable so that it could be checked at the end of the quiz. Although I did not end up implementing this functionality, I gained valuable insight into methods that I could use to try and fix my problem. For instance, I was not aware that there was a console that you could log to using JavaScript and one answer showed me this (Olsen, 2020). In my references, I have included any other questions that I posted to Stack Overflow.

This is a much less quantitative method of research when compared to the statistics I gathered from the reports and studies mentioned earlier, but it was an essential part of the development of my website.

## Secondary Research

### Secondary Research For Website Content

Secondary research was an essential part of this project, information that I retrieved using it makes up the majority of the content on my website. Therefore, it was important that the information I found using any secondary research methods was accurate and valid. This is especially important given the nature of the content.

I used the internet to collect this information for a number of reasons. I have outlined the most significant below:

* Information is quickly and readily available
* It is free
* Information from government agencies is widely available

Of course, these advantages need to be taken with a grain of salt. Anyone can post some information on the internet true or not. Therefore, as I mentioned earlier, the information had to be accurate, valid, and reliable. To minimise the likelihood of finding information that is none of these three criteria, I only looked for reports, studies and statistics from government and professional agencies. These include:

* Public Health England (PHE)
* The NHS
* The British Medical Association (BMA) board of science
* Home Office

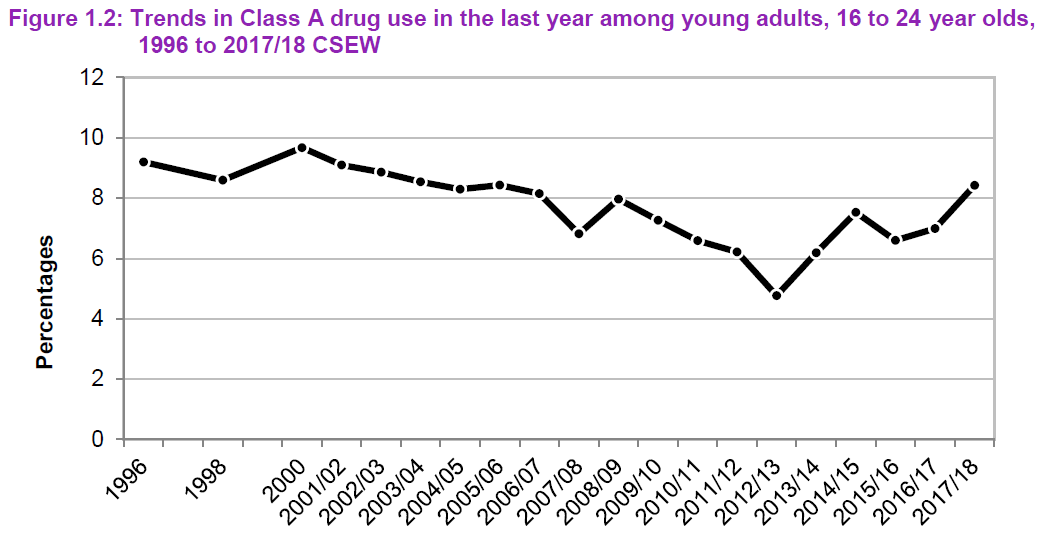
I was able to gather valuable quantitative results from these agencies that provided a lot of insight into the topic for myself and content for the website. I have included an example of a visualisation of the information I gathered, provided by the Home Office, below.

Figure - Trends in Class A drug use in the last year among young adults, 16 to 24 year olds, 1996 to 2017/18 (Home Office, 2018)

There are a few disadvantages to the approach I took, however. By limiting my scope to only government and professional agencies, I am disregarding a lot of information from other sources that could help me in the development of my website.   
This is something that I had to accept as the importance of accurate, valid, and reliable data is huge when creating a website regarding something as sensitive and important as substance abuse.

### Secondary Research During Implementation

Throughout the implementation of the website, I had to conduct more secondary research to gather information needed regarding the coding languages I used, especially PHP. I used a few different techniques to gather this information but, once again, the main technique was using the internet.

I mentioned in the Primary Research section that I had asked questions on Stack Overflow, I also used Stack Overflow to search for answers to problems I was having without asking a question myself. More often than not, I found a large amount of content on the website regarding problems I was having throughout the development in the form of previously asked questions and their corresponding answers.   
One example of this is when I was trying to work out how to navigate from one modal to the other using a button. I understood that I had to close the current and open the next but being new to JavaScript and modals, I was not sure how to approach the problem. Fortunately, I found a question regarding the problem and one of its answers helped me fix the problem perfectly (Luurtsema, 2013).

Another method I used, albeit less frequently, was video tutorials. There is a vast array of video content available online concerning PHP, JavaScript, web design and every other aspect of this project. This research method is as valid as looking through Stack Overflow for answers to my questions, and the reason that I used it much less is simply a personal preference. I feel that I understand and take in content more effectively when I am reading it rather than watching or listening.  
The most significant use of this method throughout my project was when I was implementing my quiz functionality. I found a two-part tutorial (WebTutsHD, 2010) that covered creating a PHP Quiz application and after following it I had the exact functionality that I was looking for in my quiz at the time.

## Development Tools

I used a number of tools throughout the development of this project, ranging from the IDEs I used to the software I used to draft the design of the website. However, there are some main tools that I used and would like to discuss and reflect upon. These are:

* The Integrated development environments (IDE) that I used
* cPanel
* phpMyAdmin
* MySQL databases

### IDEs

In my interim planning and investigation report, I wrote that I had decided on using Adobe Dreamweaver as my IDE for the development of my website. I initially decided on Dreamweaver because of its functionality for HTML, CSS, PHP, and JavaScript as well as its live preview feature that I believed would speed up the development process and improve the quality of the website. I quickly came to realise that the live preview feature did not work for PHP pages if you do not have a live server connection and without one, I could not use it at all. To test my code, I had to reupload the PHP file to my File Manager in cPanel and reload the page every time. Despite this, I carried on using Dreamweaver for another two months.

During this time of development, I was using Visual Studio Code for another project at university and much preferred developing in that environment. It felt quicker, more intuitive and a lot more modern. After some research, I found an extension called PHP IntelliSense (Becker, 2019) that handles auto-completion refactoring support for PHP in VS Code. It was extremely simple to move all of my code over to that IDE, which was the only thing that I was worried about, so I made the change from Dreamweaver to VS Code.  
I’m very glad that I made the change as I believe that developing in an IDE that you are comfortable in and having confidence in its ability and feature-set is a hugely important aspect of development. I think that if I decided to stay with Dreamweaver, the time it would have taken me to develop the same website would have been a lot longer.

### cPanel

cPanel is a web hosting control panel that provides users with a GUI and automation tools designed to simplify the process of hosting a website. It has a huge library of features that range from tools for databases to Git version control. I was only interested in three of these features; phpMyAdmin, MySQL Databases and the File Manager. These three tools combined had all the functionality that I would need in order to create my website.

Accessing cPanel is extremely easy as a Brighton University student as it is just a case of logging into Brighton Domains and, in my case, creating a domain. I then place all the files required for my website (HTML, PHP, JavaScript files, images, etc.) into the “public\_html” folder in the File Manager and then the website is live.   
I could then implement any database functionality that I needed for the website by creating a new database in MySQL Databases as well as a new user (in this case, myself) and giving them complete privileges. All the database design and implementation, such as running SQL queries to create tables, can then be done through phpMyAdmin.

Having all these tools so easily accessible and all in one simple and intuitive graphical user interface really helped me in efficiently developing my website.

### phpMyAdmin

phpMyAdmin is a free, open source administration tool for MySQL. It has a lot of great features for MySQL administration, including but not limited to:

* Web interface
* Import data from CSV and SQL
* Export data to a number of formats, for instance XML, PDF, Word, Excel and more
* Creating complex queries using query-by-example (QBE)
* Searching globally in a database or a subset of it

(Wikipedia, 2020)

This feature-set easily makes phpMyAdmin one of the best tools to design, create and modify MySQL databases and made it perfect for my use case. I did not have much experience in database design and implementation before this project, so in order to maximise efficiency and minimise potential problems during development using phpMyAdmin made perfect sense due to its accessibility and simplicity.

There are a number of alternatives I could have used in place of phpMyAdmin. These include Adminer (formerly phpMinAdmin), a full-featured database management tool also written in PHP. It consists of a single file ready to deploy to any target server and is one of the most popular alternatives to phpMyAdmin. Its user interface is a lot more simple than phpMyAdmin’s which, it could be argued, makes it easier to use (Adminer, 2020).   
However, once you are used to the phpMyAdmin user interface it is extremely simple to use and I never experienced any problems with it. The integration with cPanel made this an obvious choice for me and I never looked elsewhere.

### MySQL Databases

The MySQL databases cPanel tool is incredibly simple and functional. MySQL databases are necessary to run many web-apps on the internet such as content management systems or online shopping carts. They also help to run smaller features in websites, such as the quizzes in my own.

Creating a new database is as simple as choosing a name for it and clicking the “Create Database” button. In order to use the created database, a MySQL user is needed. Creating this is simple too, all that is needed is a username and password. After all this is done, you select the user and add them into the database and the database is ready to be used.

Once again, there are many alternatives to MySQL. MariaDB is probably the most popular alternative, it is a database server with drop-in replacement for MySQL created by the original developers of MySQL. It has a lot more storage options available and is faster than MySQL.  
These advantages are not significant enough for me to have considered MariaDB as an alternative, however. My storage needs are extremely small due to the simplicity of my database and because of the small number of queries that I am running speed is not really a factor. On top of this, the integration with cPanel is invaluable once again so I did not consider any alternatives to the MySQL database tool.

## Testing

### Methodology

Due to the way in which I developed my website and the fact that I was learning new techniques and content during the development, I used the unit testing approach to test my website. Unit testing is an approach to software testing where individual units, or components, of a software are tested. The purpose of this approach is to validate that each unit of the software performs as designed and intended. A unit is the smallest testable part of any software. This could be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method and this may belong to a base/super class, abstract class or derived/child class (Software Testing Fundamentals, 2020).

This suited the development of my website well because I was mostly learning one technique or functionality before moving on to the next. Testing that functionality before moving to the next felt natural from the beginning of development.   
A good example of how this style of testing was effective was during the initial development of my quizzes. As this was brand new knowledge to me, there was a big risk factor and possibility that I would not get it completely right the first time so testing it immediately and thoroughly was essential.

Unit testing increased my confidence in changing and maintaining the code I had. If I had created good unit tests and if I ran them every time I changed code, I would be able to quickly catch any problems occurring because of those changes to the code.   
This approach to testing also promotes more modular coding which helps tremendously in unit testing as well-designed modular code can drastically lower the number of possible causes of a problem.

### Test Plan

There are many tools and frameworks available for unit test almost any language. This kind of automated testing can significantly improve the workflow of development by allowing pre-written tests to drive the development process.   
Despite these tools and frameworks being available to me, I decided to create my own simple test plan and stick to that throughout the development of the project. Because I was developing in several languages, I decided to design the plan to be quite general so that it could be applied to any of the languages I was using.

I have included this test plan template and a few examples that I used during development in the appendix.

# Product Description

## Introduction

In this section, I will discuss the design, implementation, and evaluation of my project. Throughout this discussion, I hope to clearly explain my choices and how I was led to them. I will talk about the challenges I faced during the implementation, how they affected development and, if there are any, the changes that occurred because of these challenges. I will also analyse my requirements and reflect on how I succeeded, or failed, in meeting them.

As part of the Implementation section, I will discuss the code that I wrote for my project. Of course, it would not be effective to copy and paste hundreds of lines of code into this report. Instead, I will present and discuss key sections of my code base. I will decide on which sections of my code to present and discuss based one of the criteria below:

* It plays an important role in the website
* It shows progress in a new skill area
* Its functionality is important to the website

This will help me to keep the number of sections of code to just two or three. This means that I will be able to discuss, explain and reflect upon each one more than I would have if I included more than three.

## Requirements Analysis

In the Interim Planning and Investigation Report, I mentioned that the core requirements of the project would mostly be gathered from brainstorming and that I had already identified most of the requirements in the report.   
This is partly true; I had identified some requirements, but I did not present them in an effective way. I also should have discussed some potential requirements that could come up and how they would affect the development of the project, rather than just state that some could come up.

I also did not discuss the fact that I would use the MoSCoW technique to identify and prioritise these requirements. Below, I have included two tables to define the priorities and visualise the requirements.

|  |  |
| --- | --- |
| Priority | Explanation |
| Must have | Describes a requirement that must be satisfied in the final product for the product to be considered a success |
| Should have | Represents a high-priority item that should be included in the product if it is possible |
| Could have | Describes a requirement which is considered desirable but not necessary |
| Won’t have | Represents a requirement that stakeholders have agreed will not be implemented in a given release, but may considered for any future releases |

|  |  |  |
| --- | --- | --- |
| Requirement | Priority | Explanation |
| The information presented on the website is presented in an informal/casual way | Must have | This is important as one of the main goals of the project is to provide a more casual approach to this topic, which is hard to find online |
| The information presented on the website is factual | Must have | Due to the sensitivity and nature of the topic this is essential |
| Quizzes for each substance discussed to test the user on the information presented | Must have | Testing will help the users take in the information on the website |
| Quiz questions and answers will be stored in a MySQL database | Must have | This will help in creating a more dynamic website as well as provide me with experience in a new area |
| The website will present the question and answers using PHP | Must have | Same as above, also increases the expandability of the page for future development |
| The website will be responsive and mobile-friendly | Must have | Essential for modern web development |
| An admin login system where someone with credentials can modify, add or remove content to the site (a tailor-made content management system) | Should have | Will result in a much more dynamic website and allow any future development to be conducted by someone with no experience in front and back-end web development |
| There will be some features on the website implemented using JavaScript | Should have | This will increase the amount of new learning for myself, but also result in a more dynamic and modern website |
| The website will contain areas of HTML & CSS development that I have not explored before | Should have | Same as above |
| Engaging and aesthetically pleasing graphics throughout the site | Should have | This will help retain user engagement as well as help modernise the website |
| Smooth and modern transitions throughout the site | Should have | Same as above |
| Relevant links and/or contacts for users if they are struggling with substance abuse | Should have | In line with the goal of helping and educating young adults and could improve user satisfaction |
| Relevant media to help discuss or present information (different to images, such as videos or gifs) | Could have | This would help to convey information to the users in easier to understand ways and diversify content on the website |
| More interactive features to reinforce the learning theory of cognitive constructivism | Could have | Features such as drag and match statements would reinforce the learning theory I discussed in the interim report and diversify content on the website |
| Login system for users to track their scores throughout the website | Won’t have | This would increase the use of the MySQL database and in turn increase new learning for the project, but it goes against the policy I have set on anonymity for the website |

## Design

In this section, I will discuss how I ended up with the designs that I have in the final product and reflect upon the decisions that lead me to implementing those designs. These designs include the user interface and three-tier architecture of the website, as well as how I designed the MySQL database.

### User Interface Design

In the methodology section of this report, I discussed how I used primary research to decide on the design of the user interface of my website. I used a simple wireframe design, shown in Appendices 4.1 – 4.7, to help present my ideas for the UI to the respondents of the questionnaire. While creating these wireframes, I was naïve and did not judge correctly what it would take to implement the designs that I had sent out. This was exacerbated by the fact that I initially decided on not using a framework to assist in the development of the website, using a custom CSS file to handle all the classes that I would need.

As I got into the latter half of the project, I realised that I did not have enough experience to implement these designs using my own CSS file in an acceptable amount of time. It was taking me far too long to get the results that I was looking for on just the homepage.   
After discussing this problem with my project supervisor, I decided to use the Bootstrap framework to speed up the development of the project. This was a big decision and moment in the project, although there was not a lot to change in the code it meant that I would have to learn how to integrate Bootstrap and get familiar with it.

This disruption in development and subsequent lack of time left until the deadline, lead me to implementing a design that did not fully represent the wireframes that I had initially designed. At this stage of development, I feel like I had to make a choice between sacrificing the planned UI design or sacrificing the content and information on the pages. Due to the nature of the content on the website, the information took precedence in this case.  
With some more time in which I could have researched elements of the Bootstrap framework, most importantly the container, column and row classes, I believe that I would have been able to successfully implement the original designs.

In my opinion, the outcome is a good alternative to the original UI designs. Although there is less media than originally intended, I did not sacrifice the amount of information and it is still presented in an aesthetically pleasing and modern way. This is not to say that it could not be improved upon however, there is a lot of room for future development and improvement which I will discuss further in the evaluation part of this section.

### Three-Tier Architecture

I outline in the Interim Planning and Investigation Report the advantages that result from employing a three-tier software architecture. The modularising and separation of the layers of UI, business logic and data storage provided me with greater flexibility during development. This allowed me to update or modify one tier of the software without affecting the other two tiers.

The three tiers are generally known as:

* Presentation tier – the front-end layer which consists of the user interface, often a graphical one accessible through a web browser or web-based application and displays content and information to an end user
* Application tier – contains the functional logic which drives the application’s core capabilities
* Data tier – comprises of the database/data storage system and data access layer, such as MySQL, Oracle, etc.

(Jinfonet Software, Inc, 2017)

The presentation tier consists of the HTML and CSS that drives the UI of the website.  
In my case, the application tier is the functional logic behind the quizzes for each drug and anything else on the website that retrieved information from the database (the data tier) through the use of PHP. The application tier also contains the small amount of JavaScript contained on the website.  
The MySQL database is the where the website’s information is stored and retrieved from, this information is passed back to the application tier for processing (displaying the information in the quizzes).

Once I defined these tiers in my documentation and started development with them in mind, it was easy to stick to this architecture design. Beginning development in a modularity focused mindset results in code that encourages further modularity. I also believe that my development in one of my other modules, Android Application Development, helped me in this regard. I was heavily focused on modularity and an object-oriented approach to developing and I brought that into this project with me.  
Despite the fact that I was not consciously adhering to this three-tier architecture, it was undoubtably advantageous and helped in achieving a modular, object-oriented code base.

### Database Design

I wanted to briefly touch upon the design of my database, despite the simplicity of it. There are six tables, two for each quiz on the website. One houses the questions and the other the answers to those questions, both with their relevant IDs.

In my initial design draft for the database, I had all the questions and answers in one table. This design completely ignored normalisation and was, therefore, inefficient. Adhering to database normalisation is very important, no matter how simple or small the database is. It means that any modifications or additions made in the future will also be efficient. I came to realise the importance of normalising tables once I tried to implement the first version of my quiz.

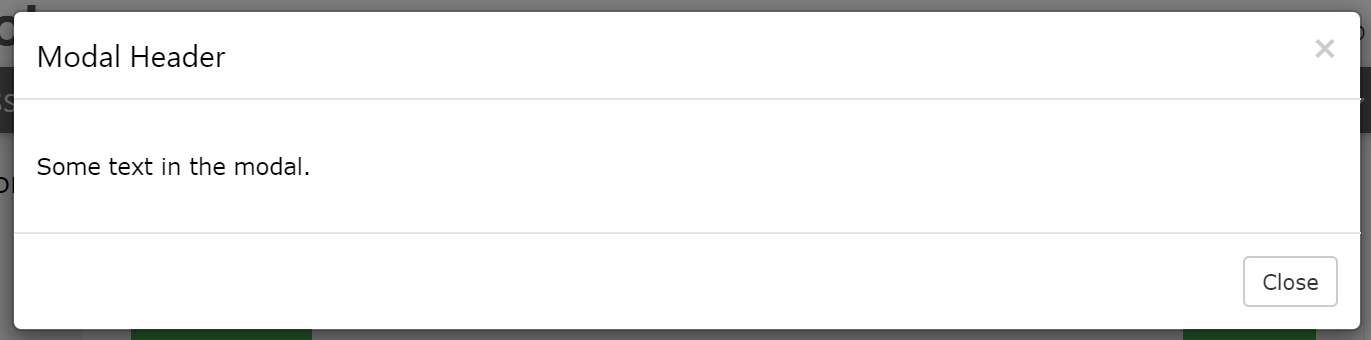
When I was first developing the functionality for my quiz, the SQL that is requires was becoming very convoluted. Normalising my database helped to reduce the complexity of the implementation and therefore the efficiency of both my database and the necessary queries. I discuss below, in the implementation section, in more detail how my quiz functionality is designed and implemented using PHP and MySQL queries.

## Implementation

In this section, I am going to discuss a few areas of the implementation of my website. I will explain the choices I made when coding these areas, as well as discuss other techniques that I could have possibly used and challenges I experience while implementing them.

### Modals

In my first designs for the quiz functionality, the user would have to press a button on the corresponding drug’s page that would take them to the first question of the quiz. After answering each question, the user would have to go to another page where the next part of the functionality is dealt with. This was inefficient and unnecessary, and I wanted a way to present the quiz to the user without them having to visit several pages or even leave the page at all. The answer to this problem was to use modals.

I was made aware of modals by my project supervisor and, therefore, had no experience in developing them. My first step was to do some research and find a tutorial on how to develop and implement modals using HTML and JavaScript. I quickly found the w3schools tutorial (w3schools.com, 2020) which guided me on implementing a modal using Bootstrap.  
Firstly, I copied and pasted the code provided on the w3schools page into one of the HTML documents that I was using to test potential HTML features for the final product. This was a basic outline of a modal, consisting of a button that would trigger the opening of the modal, the container for the modal and the content for the modal itself. The content of the modal consisted of a header, title, body, and footer. This code, found in appendix 5, resulted in the modal shown below.

I liked this implementation, however I wanted to have a greater degree of control over the different elements of the modal. I found another w3schools tutorial (w3schools, 2020) on implementing a modal using CSS and JavaScript. Not only did this give me a greater degree of control, it was also a chance for me to learn and implement some JavaScript, which I had not done before. This method was quite different to the previous example, so below I will explain how I developed the modal in my final product.

#### HTML

The HTML for this version of the modal implementation was relatively unchanged to that shown in appendix 5.1. Two changes that I need to identify however, is the exclusions of data-toggle, data-target and the modal fade class.   
data-toggle="modal" and data-target="#myModal" were used in the buttons to open and close the modal. This functionality would now be handled by the JavaScript that I implemented.  
The removal of “class="modal fade"” from the modal division is more of a change in CSS, so I will cover this in the next section.

#### CSS

In the first implementation, I had none of my own CSS integrated with the modal (Bootstrap handled all of it). To have more control over the colours and other visual aspects of the modal, this implementation relies more on custom CSS.

Firstly, there is the “*modal*” class that acts as the container for the modal content. In this case, it takes up the full screen and uses a black background colour with a set opacity to create the backdrop for the content. It is hidden by default, so as not to show a block on the page when it is not needed. When the button to open it is pressed, the display mode is set to block, this is handled by the JavaScript on the page so I will discuss it in the next section. The code for this class can be seen in appendix 5.2.

The other significant CSS class for the modals, is the ­“*modal-content*” class. Whereas the “*modal*” class’ position is set to fixed, it is set to relative in “*modal-content*” so that the content will be easily seen no matter the position on the page or size of the window. In this class, shown in appendix 5.3, I define the number of pixels from the top that it should appear at 500 pixels. This proved to be the best number when testing with different window sizes.   
As there is only one “*modal-content*” class, there is one set width for every modal no matter what content is inside of it. In future development, I could create a few different classes with different sizes that would correspond to a certain type of content.  
It must be said however, that at smaller resolutions such as those typically found in phones with smaller screens the modal tends to get “stuck” to the bottom of the screen because of this 500-pixel rule. This is something that I will try to fix in the future development of the website.

One other part of the CSS in this version of the modal I want to talk about is the animation used to display the modal after the button is pressed. It is a simple implementation but very effective and aesthetically pleasing. As shown below, I defined an animation named *animateFadeIn* that fades an element in by increasing the opacity of said element from 0 to 1. Originally, it slid and faded in from the left because I had the “*left*” part defined as -1000px, so way off the screen. I thought this was too excessive, so I changed it to simply fade in from the centre by using 0px for both parts of the animation.

/\* Add Animation \*/

@keyframes animateFadeIn {

  from {

    left: 0px;

    opacity: 0;

  }

  to {

    left: 0px;

    opacity: 1;

  }

}

#### JavaScript

I have included the entirety of this script in appendix 5.4 but I will reference smaller parts of it throughout this explanation.

Incorporating JavaScript into the modal increased the expandability of the feature and allowed me to declutter my HTML. As I mentioned earlier, it also gave me a lot more control over the functionality of the feature as well.   
This script contains three main functions, these are:

* Displaying the modal after a button is clicked
* Closing the modal when the cross is clicked
* Closing the modal when the user clicks anywhere outside of the “*modal-content”* division

To perform all these functions, I had to define some variables that corresponded to the necessary elements on the page, such as the close buttons. The code below shows how simple it is to do this:

var variable = document.getElementById(elementID);

Where “*variable*” is a user defined name for the variable and “*elementID”* is the ID of the element intended to be assigned to the variable. I did this for three open buttons, close buttons, and modals. Three because I was planning to have three questions, one on each modal.

***Displaying the modal after a button is clicked***To do this, I assigned three variables (“*btn*”, *“btn2*”*,* “*btn3*”) to the corresponding buttons. Then I created three functions, one for each button, that are called when the button is clicked. In the first function, for the “*btn*” variable, I set the display style for the matching modal to “*block*” rather than the aforementioned “*none*”.   
The functions for the next two button variables are the same, however they also “close” the last modal by resetting the displayer style to “*none*”.

        btn2.onclick = function () {

          modal.style.display = "none";

          modal2.style.display = "block";

        };

***Closing the modal when the cross is clicked***This uses the same technique as closing the previous modal in the functions I described above. The only difference is that is has an on click listener for the variables assigned to the close buttons on each modal.

***Closing the modal when the user clicks anywhere outside of the “modal-content” division***To do this, I used an if statement with two else if statements to identify what the user is clicking on. If the user clicks on the element corresponding to the variable *modal, modal2* or *modal3,* then the display style of that element would be set to “*none*”.  
It is important to note that these functions work as the “*modal*” class is everything on screen but the class “*modal-content*”, which is what the user actually sees as “the modal”.

#### Results

This resulted in a nice looking and fluid implementation of the quiz, an example of which I included in appendix 5.5. I tried to develop functionality that would check what answer the user had chosen and store it as a JavaScript variable before moving on to the next question. I could then use these variables, after the user clicked a “Submit answer” button, to check their answers and display to them how many they got correct.

I attempted to implement this functionality for a few weeks, but I could not get it working properly. I had to make the decision to give up on developing it as it was taking up far too much time and was causing me to fall behind in other parts of the project. I repurposed the modals and they now, on some parts of the website, display extra information to the user if they click the button.  
I intend to go back to this functionality in future development, as it is the best possible implementation of the quiz and would help in achieving the modern and fluid aesthetic that I had aimed for at the start of the project.

### The Quiz

#### Connecting to my database

At the heart of the functionality of the quiz, there is the code that handles the connection to the MySQL database. This was the first PHP that I developed for the project and, as such, I followed a tutorial that I found on w3schools (w3schools, 2020) for some help.

This made the implementation relatively simple, especially as the first example uses an object-oriented approach. I defined four string variables containing:

* The database host address, in my case “localhost”
* My username for the database
* My password for that user
* The database name

Another variable, *$conn*, is defined as a new *mysqli* with the four strings above passed as arguments. *mysqli* represents a connection between PHP and a MySQL database. Having all of the code that I described above in one function, *OpenCon()*, means that I can call this function (as well as one to close the connection) in any of my other PHP files. All I need to include in those other files is one line of code, shown below:

include 'db\_connection.php';

The entirety of *db\_connection.php* can be found in appendix 6.1.

#### Running an SQL Select statement using PHP

This is a multi-step process which, at first, was a little daunting when I looked at it in whole. Once again, I was using a w3schools tutorial that explained how to run a select statement on a table using PHP (w3schools, 2020). I quickly made a test table in my database using phpMyAdmin that contained three columns: an ID, a first name and a last name. I populated this table with four rows of names to use in my testing.

Firstly, I defined a string variable containing the SQL Select statement that would retrieve all the information from the test table.

$test = "SELECT \* FROM test\_table;

This is then ran and assigned to a variable called *$result.*

$result = $conn->query($test);

If there are more than zero rows returned from this query, the function *fetch\_assoc()* puts all the results into an associative array that can then be looped through. The *while()* loop loops through the result set and outputs the data.

while($row = mysqli\_fetch\_assoc($result)) {

  echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "<br>";

}

#### Using SQL queries to display the answers and questions of the quiz

After testing that the method I laid out above worked and that I understood it properly, I set out to display my questions and their corresponding answers in the same way. I tried to make it as simple to display the text as possible when writing my SQL queries, however I had proactively made it a lot easier for myself by normalising my database, as I discussed earlier.  
To help make this explanation a lot clearer, I included an example of how I displayed one of my questions and its answers in appendix 6.2.

The main difference in my implementation for the quiz when compared to the tutorial that I followed is how I used *echo*. Using the technique from the tutorial, the HTML needed is written using *echo*. This is fine for smaller implementations like the tutorial, but when more complex HTML is needed it can quickly become convoluted.   
To counteract this, I dissected the PHP while loop into two separate blocks of PHP and wrote the HTML in between them. This made it a lot simpler for me as all I had to do was echo the column “*answer\_text*” inside the HTML *label* tags.

Once I got this working for one question and its set of answers, it was really simple to expand and implement all of the other questions and answers as all that needed changing was the variable that *$row* is set to in the argument of the while loop.

#### Submitting the selected radio inputs

It would not be much of a quiz if the user could not submit the answers they gave and received some feedback on it, so these radio buttons were contained within a form. This form’s action (the destination URL) in this example was a PHP page named *marijuana\_dest.php*. This is where I handled the processing of the user’s answers. I have included the code for the functionality that I will explain below, in appendix 6.3.

Firstly, I assign a variable to the posted answer for a specific radio input name. These names are unique to each question, for example “question1\_choices”, “question2\_choices”, etc. The answers each then have a unique ID (1 – 3). I use *$\_POST[‘input name’]* to retrieve the values for the user’s answer. *$\_POST* returns an associated array of variables to the current script via the HTTP POST method (PHP, 2020). In this case, it will return the unique ID for the answer that the user chose.  
I then check this against the correct answer using an if statement, and if it satisfies the condition then the variable *$totalCorrect* is incremented by one.

At the moment, I just compare the variable to 1, 2 or 3. A better way to check whether the user’s answer is correct would be to include a column in the tables holding the answers called *is\_correct.* This would be an enum column with either a 0 (incorrect) or 1 (correct) for the corresponding *answer\_id*. I could then use another SQL statement to check whether the unique ID returned from the post method matches against the *answer\_id* where ­*is\_correct* is equal to 1.  
I did not implement this due to time constraints and it is something I will definitely include in my future development.

#### Giving feedback on the scores

I wanted the destination pages to display which score the user got and then some feedback on that score. To do this, I needed to check what score the user got and then display the corresponding content. I used the same technique as I did in displaying the questions and answers to achieve this. I wrapped HTML that displays the content in a PHP if loop that checked what the variable *$totalCorrect* is. I have included an example of this in appendix 6.4 which shows what content is displayed when the user gets 0 answers correct.

## Evaluation

Overall, I am happy with the choices I made during the development of the project and the outcome of these choices. I believe that I met all of the requirements that I deemed essential to the project and created a solid code base to achieve the requirements that I did not meet in this first version in future development. I believe that my design choices (be it for the user interface, database or system architecture) helped me tremendously in efficient implementing the website. The choices that I made throughout the implementation, such as sacrificing the use of modals for the quiz, allowed me to deliver a completed end product and fulfil the most important goals I had for the project.

This is not to say that I am completely happy with the outcomes of these choices, however. There are a number of areas that I could have been more efficient in developing or, if I had managed my time between all of my modules better, spent more time developing and therefore resulted in a better product. Other choices I made, such as not researching more possible uses of JavaScript for my website, would have increased the amount of new learning and, most probably, result in a website with more functionality.

# Critical Review

## Risk Analysis

In my interim planning and investigation report, I identified several acute and chronic risks that could present themselves throughout the development of the project, along with their likelihoods and consequences. In this section, I will revisit those risks and discuss them, as well as discuss risks that occurred that I did not foresee.

### Acute risks

These risks can also be regarded as event-driven risks, caused by individual events. I had initially identified six possible risks:

* My supervisor leaves the university
* My laptop breaks
* I lose my backup
* I become ill
* University strikes
* Brighton domains/cPanel breaks or corrupts my data

### Chronic risks

These risks can also be regarded as evolving risks; they grow over time before coming to a head. I had initially identified five possible risks:

* Underestimating how long a task in a sprint will take
* Underestimating how long a sprint will take
* Unforeseen bugs continuously break code
* Database design is not functional
* Poor UI/UX design throughout

I was quite naïve when identifying these risks and ignorant to the large number of other acute and chronic risks that could occur throughout the development of my project. Although my intention was not to list every single possible risk, I wanted to cover those that I thought were most likely.   
What I ended up actually doing was identifying risks that represented different levels of consequence or likelihood just to show that there were different levels.   
For instance, my supervisor leaving the university was extremely unlikely and I think that I only included it in order to provide a risk that was highly unlikely but would have severe consequences.   
Similarly, underestimating how long a sprint will take in a project full of brand-new learning is a lot more likely than I had initially identified but the consequence a lot lower than I identified. Rescheduling my time spent in other sprints to apply more time to the problematic sprint would be a quick and easy fix in this situation.

Although my initial identifications of these risks and the lack of identification of other risks did not directly affect the development of the project, I believe that if I had improved these identifications I would have been more aware of when the risks were occurring and that I would have been better to deal with them.

### COVID-19 pandemic

It would not be right for me to revisit the risk analysis and not discuss the global pandemic that undoubtedly affected all students at the university. On the 23rd of March, we had moved to online teaching and the university was closed.

Personally, it meant that I had to return home to help my mum care for my brother and grandfather, who are both high-risk to the virus. This meant continuous disruptions to my studies as I would have to make sure my brother was waking up on time, eating and doing the work that his college had set him. I had to travel to my grandfather, who lives an hour away, twice a week to do shopping for him and see how he is.  
After three weeks of this, I had fallen heavily behind on all my work. After talking to my mum, we agreed that I would return to my flat in Brighton to try and catch up as much as possible.

Fortunately, I was granted a two-week extension on top of the one-week extension we had already been given. Despite this, I had to work hard to catch up on all my work and it did end up affecting the quality of my final product. Although this is not something that I could have predicted, I think that I could have handled the disruption to my work better if I had recognised the severity of the disruption earlier.

## What has been successful?

What I would deem as the most successful part of my project is the representation of progress that I showed throughout. I believe that it clearly shows the progress that I made in all the areas of new learning. There is a substantial amount of PHP in the project and, being my first experience in the language, I believe that I explored enough areas of it to understand its usefulness. It is a very powerful language that can be used in a plethora of situations, and its compatibility with MySQL databases allowed me to create a more dynamic website than I would have without it.  
Similarly, using a MySQL database for the first time and integrating it with my website showed me the true potential that a dynamic website has. There are a lot of possible features that I could, in the future, implement using a database. I discuss these more in the section below.

In terms of my whole project, I believe that it has been successful. I set out with the main goal to develop a website that provides important, factual, and informative information to the users and I think that I have achieved that. I regarded this as my most important goal, so I am proud that I achieved it.

## What could I improve upon?

There are a few areas of this project that I have identified and would like to improve in any future versions.

Firstly, I would like to increase the amount of information presented on the website. This could be in the form of a greater amount of information for the drugs, or more pages for different drugs. As I mentioned above, providing useful information to the people who visit my website is the main goal of this project and implementing more onto the website would help cement that goal succeeding.

Further integration and expansion of the MySQL database, as well as increasing how reliant the website is on the database would help me in creating a much more dynamic website. For instance, creating a content management system accessible with some admin credentials that directly linked and allowed for the modification of the information presented on the site to the database would be a huge improvement to the site and future-proof it.

I would also like to do more research into Bootstrap and JavaScript in order to improve the overall design of the website. In my interim report, I outlined several modern websites with aesthetically pleasing design features that I would like to implement in my own. This would help in attracting and engaging users of the website.

## The future

The experience I have gained in front and back-end web development has really interested me in the area. How much I have learnt in a relatively short time and how much I enjoyed the development process makes me want to return to the project in the future and implement the features that I discussed in the section above.  
I am passionate about the subject area and believe that the project is important, so any improvement to it would be beneficial, especially the potential improvements that I outlined earlier.

# References

Adminer, 2020. *Adminer: Database management in a single PHP file.* [Online]   
Available at: https://www.adminer.org/  
[Accessed 28 May 2020].

Association for Project Management, 2020. *What is agile project management?.* [Online]   
Available at: https://www.apm.org.uk/resources/find-a-resource/agile-project-management/  
[Accessed 23 May 2020].

Becker, F., 2019. *PHP IntelliSense.* [Online]   
Available at: https://marketplace.visualstudio.com/items?itemName=felixfbecker.php-intellisense  
[Accessed 26 May 2020].

BMA Board of Science, 2014. Influences on illicit drug use. In: *Drugs of dependence: the role of medical professionals.* London: s.n., pp. 61-86.

Dawson, C. W., 2015. Questionnaires. In: *Projects in Computing and Information Systems.* s.l.:Pearson, p. 32.

Department for Education, 2012. *DfE and ACPO drug advice for schools,* London: DfE.

Home Office, 2018. *Drug Misuse: Findings from the 2017/18 Crime Survey for England and Wales,* London: Home Office.

Home Office, 2019. *Drugs Misuse: Findings from the 2018/19 Crime Survey for England and Wales,* London: Home Office.

Jinfonet Software, Inc, 2017. *3-Tier Architecture: A Complete Overview.* [Online]   
Available at: https://www.jinfonet.com/resources/bi-defined/3-tier-architecture-complete-overview/  
[Accessed 13 November 2019].

Luurtsema, E., 2013. *Bootstrap modal: close current, open new.* [Online]   
Available at: https://stackoverflow.com/questions/18253972/bootstrap-modal-close-current-open-new  
[Accessed 26 May 2020].

NHS, 2018. *Statistics on drug misuse,* London: NHS.

Olsen, H., 2020. *How can I access a user's answer for my quiz on a different PHP page?.* [Online]   
Available at: https://stackoverflow.com/questions/60492083/how-can-i-access-a-users-answer-for-my-quiz-on-a-different-php-page  
[Accessed 26 May 2020].

Olsen, H., 2020. *How can I make my post action in a modal take me to another modal?.* [Online]   
Available at: https://stackoverflow.com/questions/61010010/how-can-i-make-my-post-action-in-a-modal-take-me-to-another-modal  
[Accessed 26 May 2020].

Olsen, H., 2020. *How can I store a user's score in a variable on my website using PHP and/or JavaScript?.* [Online]   
Available at: https://stackoverflow.com/questions/61614042/how-can-i-store-a-users-score-in-a-variable-on-my-website-using-php-and-or-java  
[Accessed 26 May 2020].

PHP, 2020. *$\_POST.* [Online]   
Available at: https://www.php.net/manual/en/reserved.variables.post.php  
[Accessed 3 June 2020].

Public Health England, 2017. *An evidence review of the outcomes that can be expected of drug misuse treatment in England,* London: Public Health England.

Public Health England, 2018. *Adult substance misuse statistics from the National Drug Treatment Monitoring System (NDTMS),* London: Public Health England.

Public Health England, 2018. *Young People's Statistics from the National Drug Treatment Monitoring System (NDTMS),* London: Public Health England.

Software Testing Fundamentals, 2020. *Unit Testing.* [Online]   
Available at: http://softwaretestingfundamentals.com/unit-testing/  
[Accessed 28 May 2020].

w3schools.com, 2020. *Bootstrap Modal Plugin.* [Online]   
Available at: https://www.w3schools.com/bootstrap/bootstrap\_modal.asp  
[Accessed 18 March 2020].

w3schools, 2020. *How TO - CSS/JS Modal.* [Online]   
Available at: https://www.w3schools.com/howto/howto\_css\_modals.asp  
[Accessed 19 March 2020].

w3schools, 2020. *PHP Connect to MySQL.* [Online]   
Available at: https://www.w3schools.com/php/php\_mysql\_connect.asp  
[Accessed 11 January 2020].

w3schools, 2020. *PHP MySQL Select Data.* [Online]   
Available at: https://www.w3schools.com/php/php\_mysql\_select.asp  
[Accessed 12 February 2020].

WebTutsHD, 2010. *Creating a Quiz app using PHP Part - 1.* [Online]   
Available at: https://www.youtube.com/watch?v=MbJJZ0tBuR4  
[Accessed 26 May 2020].

WebTutsHD, 2010. *PHP Quiz app Part -2.* [Online]   
Available at: https://www.youtube.com/watch?v=z-LhA3ZWZOM  
[Accessed 26 May 2020].

Wikipedia, 2020. *phpMyAdmin.* [Online]   
Available at: https://en.wikipedia.org/wiki/PhpMyAdmin  
[Accessed 28 May 2020].

Zenkit, 2018. *7 Popular Project Management Methodologies And What They’re Best Suited For.* [Online]   
Available at: https://zenkit.com/en/blog/7-popular-project-management-methodologies-and-what-theyre-best-suited-for/  
[Accessed 26 May 2020].

# Appendices

Appendix 1 – Record of Meetings With Supervisor

***27th September***

Educational experience website on drugs

Topics:

* Statistics
* Addiction
* Effects
* Common misconceptions?
* Places where you can get help
* **EVIDENCE**
* Consequences of prescribed drugs

**INTERACTIVE —>** sub sections, quizzes, click to reveal, aesthetically pleasing

* Brighton domains
* Primary research from target audience
* Investigate wireframe software for mockups

**Proposal**

* Dynamically updatable through back-end
* Mention languages
* Mention research
* Quiz - explain interactivity
* Brighton domains

***3rd October***

**On proposal**

* Age up to 18 to clear ethical issues
* How to test learning (quizzes)
* Research educational side of the project
* Educational theory
* Mention database due to dynamic functionality
* Start date

**22nd October**

* Problem area - why am I doing what I’m doing? - more efficient? etc
* Justifications of choice, back it up with research
* Statistics in report to identify problem area
* Educational policy on drugs, what age?
* Tool to “plug the gap” at Uni at this age
* Project methodology, what options, why?
* MANAGING THE PROJECT

***5th November***

Highlight problem area more

Two end users:

* Admin
* Generic
* How do people learn
* Gamification - tally score from quizzes to let users know how they did by the end of the experience (could make it quite linear?) e.g. “You answered 28 out of the 30 questions right, you were clearly paying attention."
* Statistical evidence on drug education with the addiction statistics
* Mention bootstrap

**Requirements**

* Key functional elements (unique features)

**Techniques**

* How am I going to use my tools

***4th February***

Onto making PHP page in Dreamweaver, display information on a page

Upload to localhost on Brighton domains

Get default connection stream for PHP

Then build a query and print onto screen

Error messages

**Later on**

Admin panel, so that the back-end can be updated/more questions and answers created

Bigger collection of questions/answers? Later on? Thinking ahead

Making sure questions and answers in quizzes are easily compared in database, to make simpler queries

***3rd March***

* Change ID for question choice\_ID
* Post variables from question to question
* Format of IDs could lead to less posting
* Get core functionality done before all above

***24th March***

**Javascript**

* Some people may turn JS off on their browser
* Worth discussion
* Minor issue

**Modals**

* Modals mentioned in report, tried them but couldn’t get along with them
* May go back to them if I can’t implement quiz in this form
* Try implement a simple version of them before making it complex
* Try different forms of them
* Straight from HTML5, moving from JS which **could** be a problem, unlikely though

***3rd April***

* Modals for other aspects than the quiz - necessary?
* Will users think they are entering another quiz?
* Ask colleagues
* Mention reflection in report (vs JS pop up boxes, why I chose modals over them)
* Advantages/disadvantages of swapping over e.g. Time, functionality, future use
* Above is new learning
* Dynamic pages - if I can't implement myself include code examples, generic that would be applicable for the next version of the website

**Testing**

* Message rest of team for testing each others project

***5th May***

* Modals fix everything
* Don't have to submit form in order to check answer, do it in the footer of modals
* Don't have to leave the page at all
* Cleaner, more efficient code
* Better for connections
* Need to implement func that checks what radio has been selected when next modal is chosen
* Text files that store answers as an alternative
* SQL queries that insert answers into database as an alternative
* Mention in report with justification why I chose what I chose

***11th May***

* Keep documenting
* Talk about the console logging, modal problem
* One of the biggest problems in this project
* Technical research - solutions I didn't know exist and trying to learn them

***19th May***

Mention to Jarod:

* If I can't get modals working use separate pages for quizzes like original due to time constraints

* Keep evidence of code for modal development
* Sample code
* Alternative method could be shown in technical guide
* Version 2

* Using modals for another part of the website, extra information? Keeping quizzes on separate pages
* Use one page with 3 questions for each quiz page

***26th May***

* In video, refer back to design and implementation while discussing/presenting a certain feature/what is on screen
* Talk about plan at start with planning documentation at the start
* Test documentation

## Appendix 2.1 – Test Plan Template

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case Number | Test Case Title | Test Case Description | Test Date |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case Number | Expected Results | Actual Results | Notes |
|  |  |  |  |

## Appendix 2.2 – Test Plan Example 1

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case Number | Test Case Title | Test Case Description | Test Date |
| 8 | Opening and closing modals | Testing the buttons that are meant to open and close modals using JavaScript | April 2nd 2020 |

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case Number | Expected Results | Actual Results | Notes |
| 8 | Modal 1 to open with button on home page, modal 2 to open with the “Next modal” button in modal 1, the same for modal 3 and all modals to close when the user clicks outside of the modal or on the X in the top right hand corner. | The buttons open up the modals correctly, however clicking outside of the modal or on the X to close the current modal only works for the first one | This is an easy fix I think, simply need to expand the JavaScript script with two more variables for the close and open buttons and two more functions to compliment these variables |

## Appendix 2.2 – Test Plan Example 2

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case Number | Test Case Title | Test Case Description | Test Date |
| 5 | Submitting answers to quiz questions | Testing the submission of the user’s answers, one by one, to the questions in the quiz | March 2nd |

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case Number | Expected Results | Actual Results | Notes |
| 5 | If the user chooses the right answer, the variable *totalCorrect* will be incremented by one. This variable is displayed at the top of the next questions page, so choosing the right answer I would expect a 1 and choosing the wrong I would expect a 0. | Same as expected results | Luckily, this worked perfectly and as expected first time. I think the use of session variables helped to achieve this |

## Appendix 2.3 – Test Plan Example 3

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case Number | Test Case Title | Test Case Description | Test Date |
| 3 | First SQL query | Initial SQL query intended to return the test table from my database | 12th February |

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case Number | Expected Results | Actual Results | Notes |
| 3 | All three complete rows of *test\_table* to be returned, with the test values that I entered in it | All fields showing as expected except the last names | After checking the PHP that I wrote to echo each line, I noticed I misspelt the column name for last name |

## Appendix 3.1 – Homepage UI Wireframe Option 1

## Appendix 3.2 – Homepage UI Wireframe Option 2

## Appendix 3.3 – Subpage UI Wireframe Option 1

## Appendix 3.4 – Subpage UI Wireframe Option 2

## Appendix 3.5 – Quiz UI Wireframe Option 1



## Appendix 3.6 – Quiz UI Wireframe Option 2

## Appendix 3.7 – Quiz Wireframe Option 3

## Appendix 4.1 – Example Modal Code from W3Schools

<!-- Trigger the modal with a button -->  
<button type="button" class="btn btn-info btn-lg" data-toggle="modal" data-target="#myModal">Open Modal</button>  
  
<!-- Modal -->  
<div id="myModal" class="modal fade" role="dialog">  
  <div class="modal-dialog">  
  
    <!-- Modal content-->  
    <div class="modal-content">  
      <div class="modal-header">  
        <button type="button" class="close" data-dismiss="modal">&times;</button>  
        <h4 class="modal-title">Modal Header</h4>  
      </div>  
      <div class="modal-body">  
        <p>Some text in the modal.</p>  
      </div>  
      <div class="modal-footer">  
        <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>  
      </div>  
    </div>  
  
  </div>  
</div>

## Appendix 4.2 – Modal Class CSS Code

.modal {

  display: none; /\* Hidden by default \*/

  position: fixed; /\* Stay in place \*/

  z-index: 1; /\* Sit on top \*/

  left: 0;

  top: 0;

  width: 100%; /\* Full width \*/

  height: 100%; /\* Full height \*/

  overflow: auto; /\* Enable scroll if needed \*/

  background-color: rgb(0, 0, 0); /\* Fallback color \*/

  background-color: rgba(0, 0, 0, 0.5); /\* Black w/ opacity \*/

}

## Appendix 4.3 – Modal-Content Class CSS Code

/\* Modal Content \*/

.modal-content {

  position: relative;

  background-color: #fefefe;

  margin: auto;

  top: 500px;

  padding: 0;

  width: 50%;

  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);

  animation-name: animateFadeIn;

  animation-duration: 1s;

}

## Appendix 4.4 – JavaScript For Modals

<script>

        // Get the modal

        var modal = document.getElementById("myModal");

        var modal2 = document.getElementById("myModal2");

        var modal3 = document.getElementById("myModal3");

        // Get the button that opens the modal

        var btn = document.getElementById("myBtn");

        var btn2 = document.getElementById("myBtn2");

        var btn3 = document.getElementById("myBtn3");

        // Get the <span> element that closes the modal

        var close1 = document.getElementById("close1");

        var close2 = document.getElementById("close2");

        var close3 = document.getElementById("close3");

        // When the user clicks on the button, open the modal

        btn.onclick = function () {

          modal.style.display = "block";

        };

        btn2.onclick = function () {

          modal.style.display = "none";

          modal2.style.display = "block";

        };

        btn3.onclick = function () {

          modal2.style.display = "none";

          modal3.style.display = "block";

        };

        // When the user clicks on (x), close the modal

        close1.onclick = function () {

          modal.style.display = "none";

        };

        close2.onclick = function () {

          modal2.style.display = "none";

        };

        close3.onclick = function () {

          modal3.style.display = "none";

        };

       // When the user clicks anywhere outside of the modal-content, close it

        window.onclick = function (event) {

          if (event.target == modal) {

            modal.style.display = "none";

          } else if (event.target == modal2) {

            modal2.style.display = "none";

          } else if (event.target == modal3) {

            modal3.style.display = "none";

          }

        };

      </script>

## Appendix 4.5 – Example of Quiz Question In Modal

## Appendix 5.1 – Database Connection PHP Code

<?php

function OpenCon()

 {

 $dbhost = "localhost";

 $dbuser = "hbo13\_olsxn";

 $dbpass = "Benjamin1996";

 $db = "hbo13\_website";

 $conn = new mysqli($dbhost, $dbuser, $dbpass,$db) or die("Connect failed: %s\n". $conn -> error);

 return $conn;

 }

function CloseCon($conn)

 {

 $conn -> close();

 }

?>

## Appendix 5.2 – Example of Displaying A Question and Its Answers through PHP

### The SQL queries

// get question text here

$questionTextSQL\_1 = "SELECT question\_text FROM marijuana\_questions WHERE question\_id ='1'";

$questionResult1 = $conn->query($questionTextSQL\_1);

// get answers here

$answerTextSQL\_1 = "SELECT answer\_text FROM marijuana\_answers WHERE answer\_id = '1' AND question\_id = '1'";

$answerResult1 = $conn->query($answerTextSQL\_1);

$answerTextSQL\_2 = "SELECT answer\_text FROM marijuana\_answers WHERE answer\_id = '2' AND question\_id = '1'";

$answerResult2 = $conn->query($answerTextSQL\_2);

$answerTextSQL\_3 = "SELECT answer\_text FROM marijuana\_answers WHERE answer\_id = '3' AND question\_id = '1'";

$answerResult3 = $conn->query($answerTextSQL\_3);

### Displaying the question as a heading

<!-- display question1 -->

    <?php while ($question1 = $questionResult1->fetch\_assoc()) {

        echo '<h3>' . $question1["question\_text"] . '</h3>'; } ?>

### Displaying the answers as labels for radio buttons

<!-- display answers -->

    <!-- answer1 -->

    <div>

<!-- open while loop here -->

    <?php while($row = $answerResult1->fetch\_assoc()) {?>

    <input type="radio" name="question1\_choices" id="question1\_choices\_A" value="1">

    <label for="question1\_choices\_A">

        <?php echo $row["answer\_text"]; ?>

    </label>

<!-- close while loop here -->

    <?php } ?>

    </div>

    <!-- answer2 -->

    <div>

   <?php while($row = $answerResult2->fetch\_assoc()) {?>

    <input type="radio" name="question1\_choices" id="question1\_choices\_B" value="2">

    <label for="question1\_choices\_B">

        <?php echo $row["answer\_text"]; ?>

    </label>

    <?php } ?>

    </div>

    <!-- answer3 -->

    <div>

    <?php while($row = $answerResult3->fetch\_assoc()) {?>

    <input type="radio" name="question1\_choices" id="question1\_choices\_C" value="3">

    <label for="question1\_choices\_C">

        <?php echo $row["answer\_text"]; ?>

    </label>

    <?php } ?>

    </div>

## Appendix 5.3 – The handling of the user’s answers in marijuana\_dest.php

$totalCorrect = 0;

// get answers from marijuana\_quiz.php

// this will return a number from 1-3

$answer1 = $\_POST['question1\_choices'];

$answer2 = $\_POST['question2\_choices'];

$answer3 = $\_POST['question3\_choices'];

// compare answers to the correct answer

if ($answer1 == 2) {

    $totalCorrect++;

}

if ($answer2 == 3) {

    $totalCorrect++;

}

if ($answer3 == 3) {

    $totalCorrect++;

}

## Appendix 5.4 – 0 answers answered correctly in a quiz code example

      <!-- if user got 0 correct -->

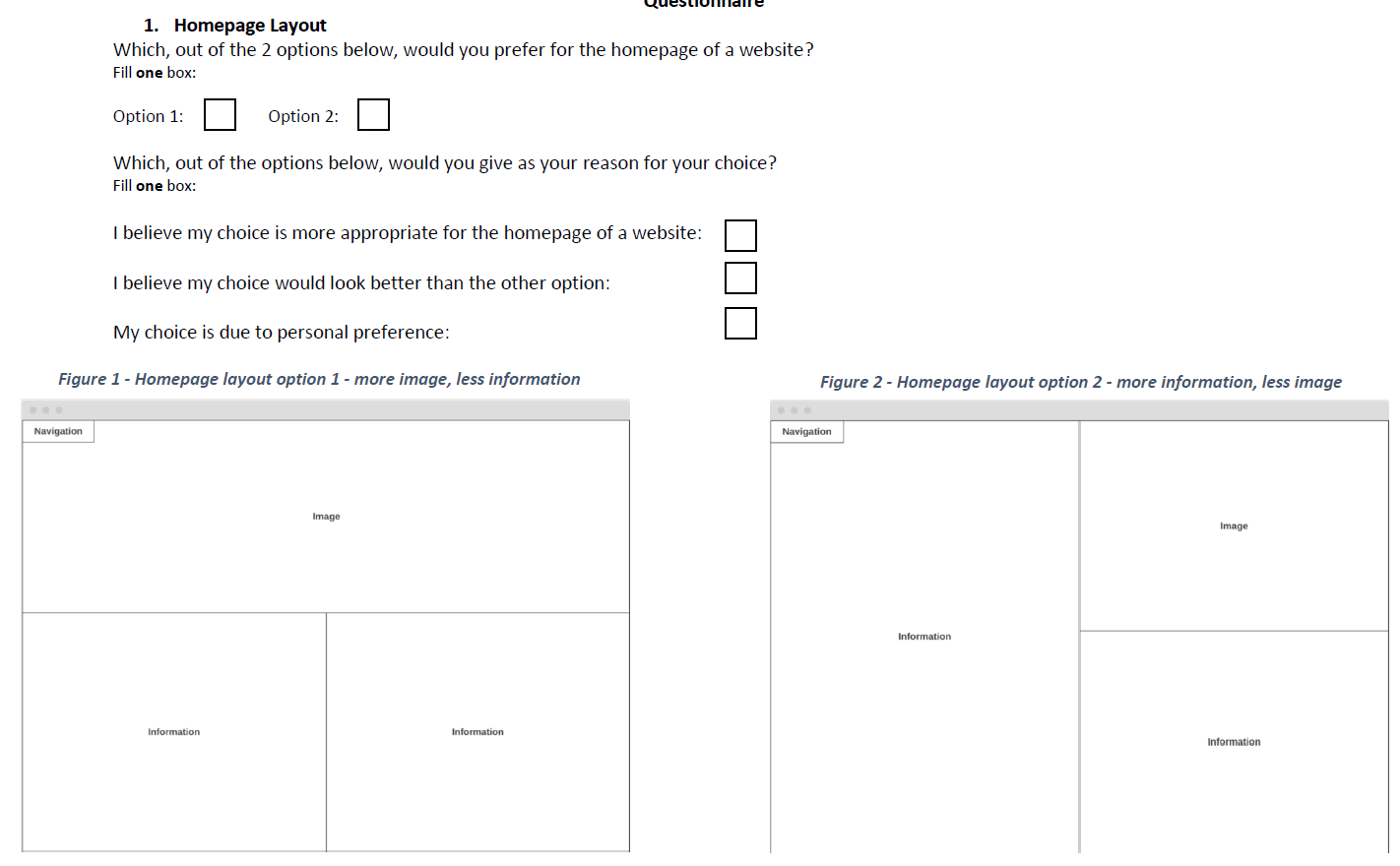
      <?php if ($totalCorrect == 0) { ?>

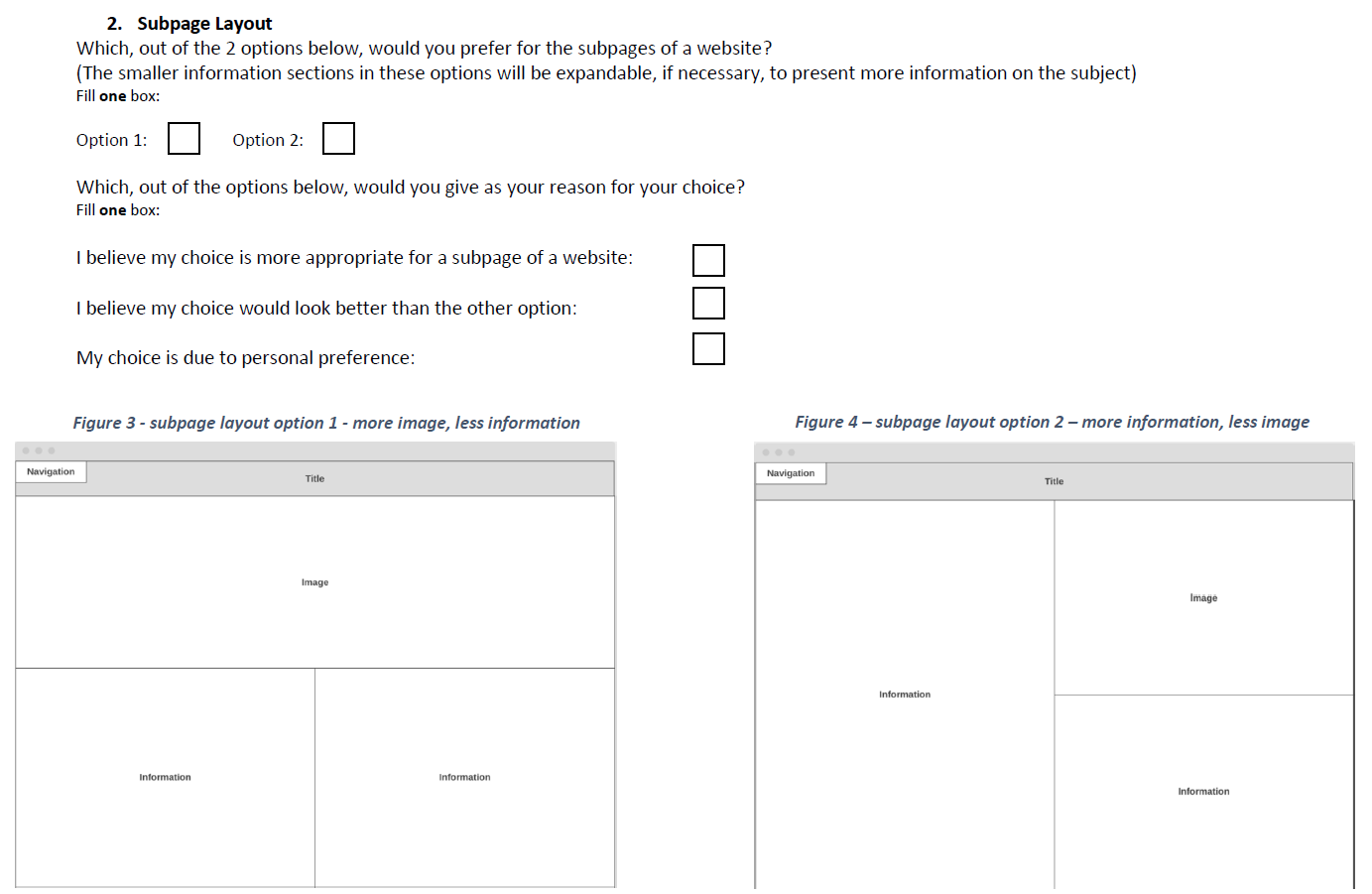
        <h1>You got 0 correct.</h1>

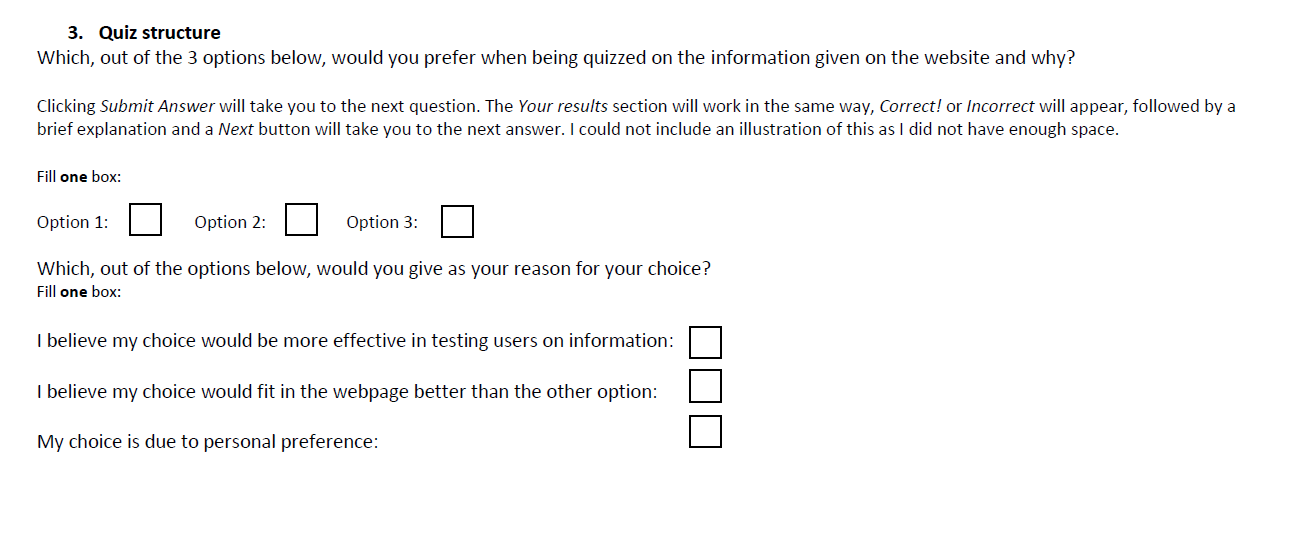
        <p>Try reading through the page again and then taking the quiz, good luck!</p>

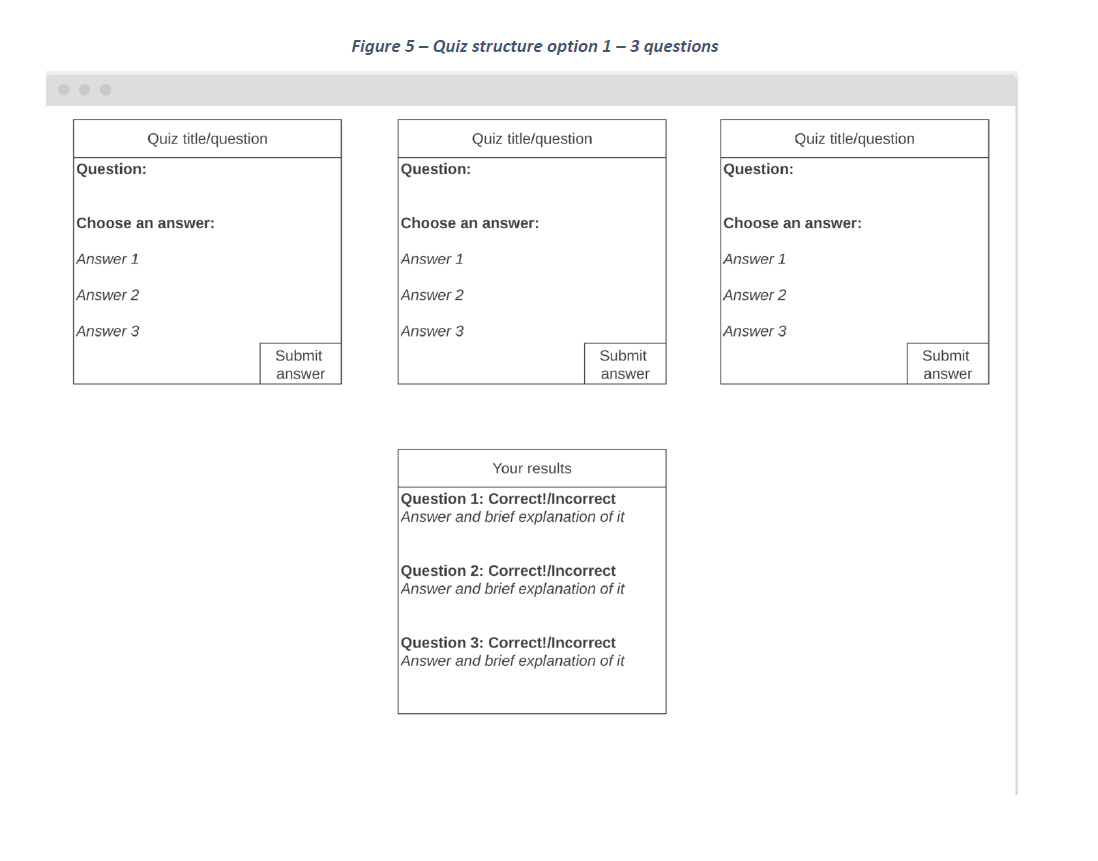
      <?php } ?>

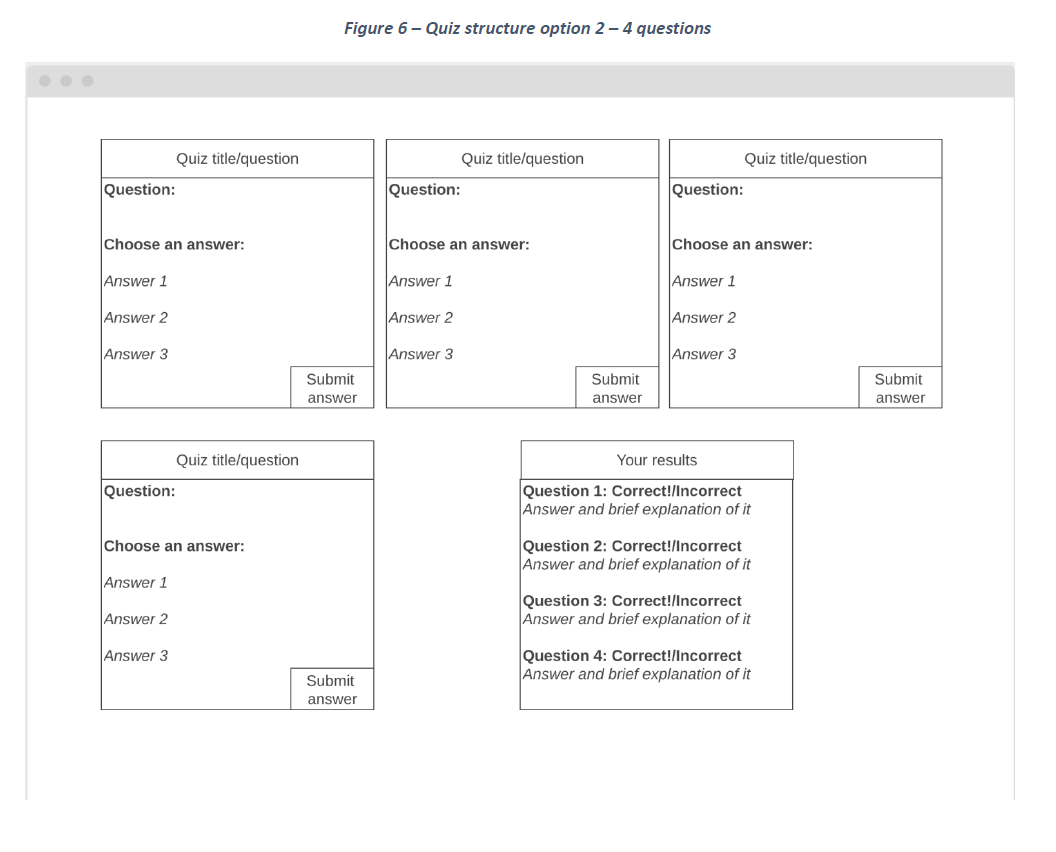
## Appendix 6.1 – Questionnaire Template

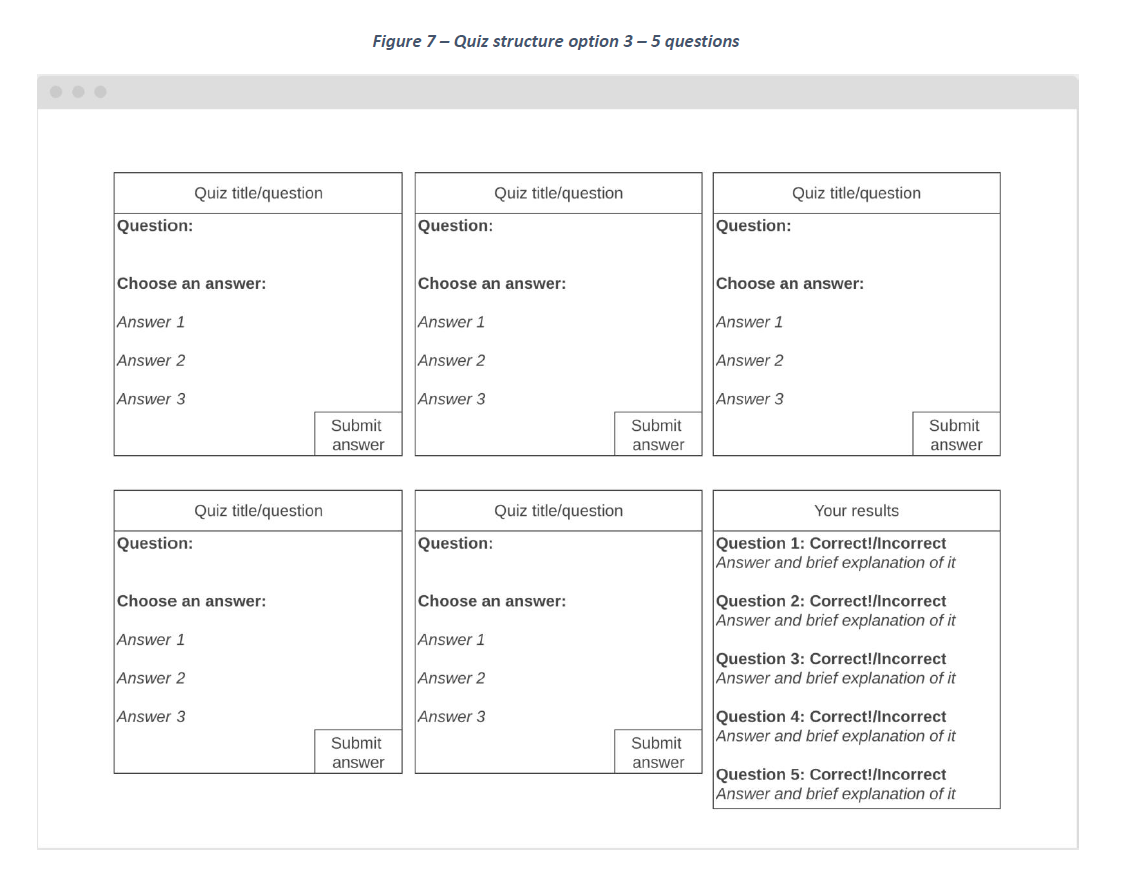












## Appendix 6.2 – Questionnaire results

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Respondent Number | Question 1 Answer | Reasoning | Question 2 Answer | Reasoning | Question 3 Answer | Reasoning |
| 1 | 1 | 3 | 2 | 3 | 1 | 2 |
| 2 | 1 | 3 | 1 | 2 | 1 | 2 |
| 3 | 2 | 2 | 2 | 3 | 1 | 2 |
| 4 | 1 | 3 | 2 | 2 | 1 | 1 |
| 5 | 1 | 2 | 2 | 2 | 1 | 2 |
| 6 | 2 | 3 | 1 | 2 | 2 | 2 |
| 7 | 1 | 1 | 1 | 3 | 2 | 2 |
| 8 | 1 | 3 | 2 | 3 | 1 | 3 |
| 9 | 1 | 2 | 1 | 2 | 2 | 2 |
| 10 | 1 | 2 | 2 | 2 | 1 | 2 |