Primary Challenge: Teachers Level Identifying Game for Primary School Aged Students

CS39440 Major Project Report

Author: Rhodri Smith ([rhs24@aber.ac.uk](mailto:rhs24@aber.ac.uk))

Supervisor: Dr Edel Sherratt (eds@aber.ac.uk)

30th April 2020

Version 1.0 (Final)

This report is submitted as partial fulfilment of a BSc degree in  
Computer Science (G400)

Department of Computer Science

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Wales, UK

Declaration of originality

I confirm that:

* This submission is my own work, except where clearly indicated.
* I understand that there are severe penalties for Unacceptable Academic Practice, which can lead to loss of marks or even the withholding of a degree.
* I have read the regulations on Unacceptable Academic Practice from the University’s Academic Registry (AR) and the relevant sections of the current Student Handbook of the Department of Computer Science.
* In submitting this work, I understand and agree to abide by the University’s regulations governing these issues.

Name: Rhodri Smith

Date: 22/04/2020

Consent to share this work

By including my name below, I hereby agree to this project's report and technical work being made available to other students and academic staff of the Aberystwyth Computer Science Department.

Name: Rhodri Smith

Date: 22/04/2020

Acknowledgements

I am grateful to friends and family for mental and financial support, not just within this project but through the 3 years of university so far. I am also grateful to the tutors and lecturers that have supported me

I’d like to thank the following staff members within the university for helping me through my time in the University and with this project. These members being Dr Thomas Jansen, my personal tutor and Dr Edel Sherratt, my project supervisor.

Abstract

In this document, we investigate the design and implement of a web-based game, named Primary Challenge, that allows for different interaction based on the type of user. We designed Primary Challenge for primary school students which records the students score on a central database, allowing for a staff member to adjust the difficulty of the game to meet the students level and to challenge them further. Primary Challenge also consists of a guest mode and access control.

Primary Challenge runs of PHP 7.2.30, JavaScipt 1.8.5, HTML 5 and CSS 3 two common web languages. Primary Challenge was developed in Atom,[[1]](#footnote-2) a free opensource IDE which is developed by GitHub.

The website can be found here;

https://users.aber.ac.uk/rhs24/MMP/LoginPage/index.php

The link to the GitHub repository;

https://github.com/Rhod4/MMP

Contents

[1. Background, Analysis & Process 8](#_Toc40236793)

[1.1. Background 8](#_Toc40236794)

[1.1.1. Focus group 8](#_Toc40236795)

[1.1.2. Languages 8](#_Toc40236796)

[1.1.3. The Game 8](#_Toc40236797)

[1.1.4. Tables 9](#_Toc40236798)

[1.1.5. Database 10](#_Toc40236799)

[1.1.6. Design ideas 11](#_Toc40236800)

[1.1.7. Other sites 12](#_Toc40236801)

[1.2. Analysis 14](#_Toc40236802)

[1.2.1. Aims and objectives 14](#_Toc40236803)

[1.2.2. Issues and Approach 14](#_Toc40236804)

[1.3. Process 17](#_Toc40236805)

[2. Design 20](#_Toc40236806)

[2.1. Visual look at the website 20](#_Toc40236807)

[2.1.1. UML and other diagrams 22](#_Toc40236808)

[2.2. Detailed Design 27](#_Toc40236809)

[2.2.1. Login page 27](#_Toc40236810)

[2.2.2. Game Page 29](#_Toc40236811)

[2.2.3. Staff and Admin Page 32](#_Toc40236812)

[2.2.4. Logout function 33](#_Toc40236813)

[2.2.5. Database connection 33](#_Toc40236814)

[2.3. Other Relevant Sections 34](#_Toc40236815)

[2.3.1. Security 34](#_Toc40236816)

[3. Implementation 35](#_Toc40236817)

[3.1.1. The Staff Page Table 35](#_Toc40236818)

[4. Testing 36](#_Toc40236819)

[4.1. Overall Approach to Testing 36](#_Toc40236820)

[4.2. User Testing 36](#_Toc40236821)

[4.3. Manual testing 39](#_Toc40236822)

[5. Critical Evaluation 40](#_Toc40236823)

[6. Annotated Bibliography 42](#_Toc40236824)

[7. Appendices 45](#_Toc40236825)

[7.1. Tests 45](#_Toc40236826)

[7.2. Diary 57](#_Toc40236827)

[7.3. Design and Prototype 59](#_Toc40236828)

[7.4. Project Goals Document 62](#_Toc40236829)

[A. Third-Party Code and Libraries 66](#_Toc40236830)

[B. Ethics Submission 68](#_Toc40236831)

[C. Code Samples 71](#_Toc40236832)

# Background, Analysis & Process

## Background

In this section, we will explore background research that was required for the project such as the choice of language, the game and security. We will also discuss third party frameworks that were utilised and our justifications for using them.

### Focus group

#### Students

The project is aimed towards young people, towards the age of primary school students. The aim of the project is to help those primary school students to improve their learning skills and development throughout school. This meant that research is needed to be carried out into what would appeal to young people or what they disliked. Research was completed through investigating other created websites and considering other people’s opinion regarding Primary Challenge.

#### Staff

The project was also aimed in a way that teachers will be able to view their students’ progress and track how they were getting on. This also meant that research needed to be completed on how to create a simple yet attractive site to teachers. The site would need to be easily used and requires little to no training on how to work the website

### Languages

There is an array of languages that powers today's web. Some of these are legacy languages such as PHP [1] to more modern languages such as WebAssembly that launched in 2017 [2].

We reviewed the use of PHP, NodeJS and WebAssembly as our primary languages for server-side languages; however, we selected PHP because of its vast browser capabilities and the immense amount of online help/discussions of other users using of the language.

We did not select Node.js as a primary language because of the developers having a lower knowledge of the language. Node.JS was also not selected due to it being a more complex to use than PHP [3]. We deemed WebAssembly as inappropriate for the project because of how new it is as a language. It also has a lot less documentation on it online, which can be attributed to be a newer language. WebAssembily allows for multiple languages to run simultaneously at near native speeds. WebAssembily is not designed to replace JavaScript, however it is designed to work with JavaScript to increase its efficiency [4].

### The Game

Creating the game was a large area of research, how and what sort of game it was going to be. During this phase of the project a few different areas to do this were investigated. One area was creating a canvas game within JavaScript and another was to use a framework that uses a canvas and WebGL. This would have been phaser [5], which is free to use software that works within the browser. The final area that was investigated was to use just basic HTML to create basic games. Using just html would lower the amount of animation that could be used within the game, however.

The canvas element that could have been used for the game would have allowed a dynamic 2d shapes. It is not designed specifically around creating games, however it is possible to make them within the canvas.

Research was also looked into how the questions and answers would be saved on the page, would they be stored on the database and pulled into an array for the JavaScript to run, or would the page randomly create the questions? Storing the questions would require a table to store them all and a facility to upload new questions. Randomly making them would not require this, but would require the creation of random questions and answers that are appropriate, e.g. incorrect answers need to be realistic to the real answer.

A multi canvas approach was looked into for this project, since detection of the canvas itself is easily possible and could allow for some sort of visually stimulating affect for when the user clicks on an answer. Issues again with this is scalability for different resolutions.

We researched different game types for the project, e.g. button clicker games or a typing sort of game. Research was done into how these games will be played and the enjoyability of those games for young people. If the games where not enjoyable, a young person is not going to be willing to play those games or may not take the games serious enough. Through this research, a retired teacher was able to give advice about what young people would and not enjoy, such as the colours of the application. If the game contained bland and boring colours, a young person may not find it visually stimulating enough to want to engage in.

Time playing the game was considered for the players score. For example if a user took 5 seconds to answer a question, should they get more score than someone who took 10 seconds. Another option with time was to limit the time a user was given with each section, e.g. if the user does not click an answer before 10 seconds, it moves onto the next question.

### Tables

Tables are an important part of the website which allow the display of ordered lists and allow interaction with data. For this tables we utilised DataTable [6] which allows for a more advanced way for the user to interact with tables. Data Tables is a free to use plugin that uses jQuery JavaScript to create events within the website by overlaying a current table with a “skin”. DataTables includes a search feature and a feature that allowed for multiple pages that a user can view.

Research was also completed into the different styles of tables, this research was carried out from the non DataTables point of view, as with the script, it would not be as necessary.

The tables need to give the user a clear view of the details within the system. If the table is too small or the text inside too small to read, the table would not be accessible to everyone, especially people with visual impairments. Using the table must also be easy, it must be easy for a staff member or admin to easily access student data as the table shows hundreds of students on the page, it will result in poor user experience.

Changing the data within the table was another area that needed to be reviewed, since a user could have issues with items such as forgetting a password, or the student’s difficulty needed to be changed. How would the user change the difficulty of the user through the website, would a pop-up box appear or could all the user’s data within the table be stored as forms and the user can change them by just clicking on it. The ability to delete users was also considered. Admins can remove users through a pop-up box that is present on the edit page.

Security is a critical item that all web-based products must consider. Hackers often attack websites, compromising their websites which can lead to data breaches. Security vulnerabilities can be found in many places, from the basic core design of the website to poor implementing of parts of the websites. Common security holes include the lack of input sanitisation that allow users to put malicious code in. Newer attacks on websites look at flaws in the design as we can we see with attacks such as Zoombombing [7], which is where a person enters a zoom call causing disruption.

One of the significant attacks that we identified is Cross-Site Scripting or XSS. XSS is a problem on the website because it could allow people who should not obtain certain data to access the data. This attack allows the attacker to latch onto another user’s session and fool the website into thinking that they are that user. This attack is common on poorly secured websites [8]

Another possible attack of this website is attacking the database. This can lead to problems for the user because it opens up the user to identity theft. Thousands of websites have been breached, many of which include their password. As of the 7th of May 2020, HaveIBeenPwned, a website that tracks leaked emails and passwords, has over 443 "pwned" websites. Many users reuse their email and passwords together, which open them up to brute force on other websites [9]. To protect users against brute, forcing on other websites, we hashed our passwords. Hashing is a one-way process which is very difficult to reverse as it sets each letter within a string, into random characters. The one-way process means that it cannot be undone, there is no method that can un-hash a string, making it a very strong command to use to encrypt data. Passwords are something that should be hashed, storing plane text passwords on a database is highly advised against [10]

After using the hash function to hash the passwords, the password\_verify [11] must be used to login with. This verifies the password to the hash that is stored within the database. This needs the passwords to be already hashed on the database to work.

We also research the different types of security issues a website could have, e.g. SQL injections, and the ways a developer could stop these from happening. If a site gets SQL injected, someone could easily steal data from the database which they should not be able to access.

### Database

Databases are an integral part of IT systems. Databases are the primary mechanism of storing data on a computer in an orderly fashion. We investigated two types of databases, which were PostgreSQL and MySQL. These are both popular database management systems and both are free to use (MySQL under the terms of the ‘GNU General Public License’ [12]). PostgreSQL is more advanced with its datatypes, allowing support of arrays and user defined types while MySQL only allows standard types [4].

We considered the advantages and disadvantages of using PostgreSQL and MySQL. We also considered the layout of the databases and how to structure user data. For example, would schools want separate database, resulting in whether the system should see all the tables or a subset of tables?

Looking at using the database when deployed, using command line was an option, however research showed that there were better methods such as the GUI interface in MySQL called phpMyAdmin [13]. phpMyAdmin allows for easy maintenance of use of the database without having to have the knowledge of the command line commands. phpMyAdmin also allows a user to control the website remotely, whereas command line would not allow remote access unless users were using a remote accessing software such as TeamViewer [14] to control the database off site. The use of TeamViewer is not user friendly and due to its centralized approach, opens itself up for a man in the middle attack.

### Design ideas

Before starting the project, a few design ideas were considered, such as how the user would transition from one screen to another. We considered a large array of design requirements, from the navigation and core design of the website down to the colour pallet that was used. We considered whether the user could navigate the website freely or only with guided access from a teacher.

The colours that were used on the website was also investigated. Too many bright colours may lead to eye fatigue, however too many dark colours would make it difficult for the user to view in dark situations.

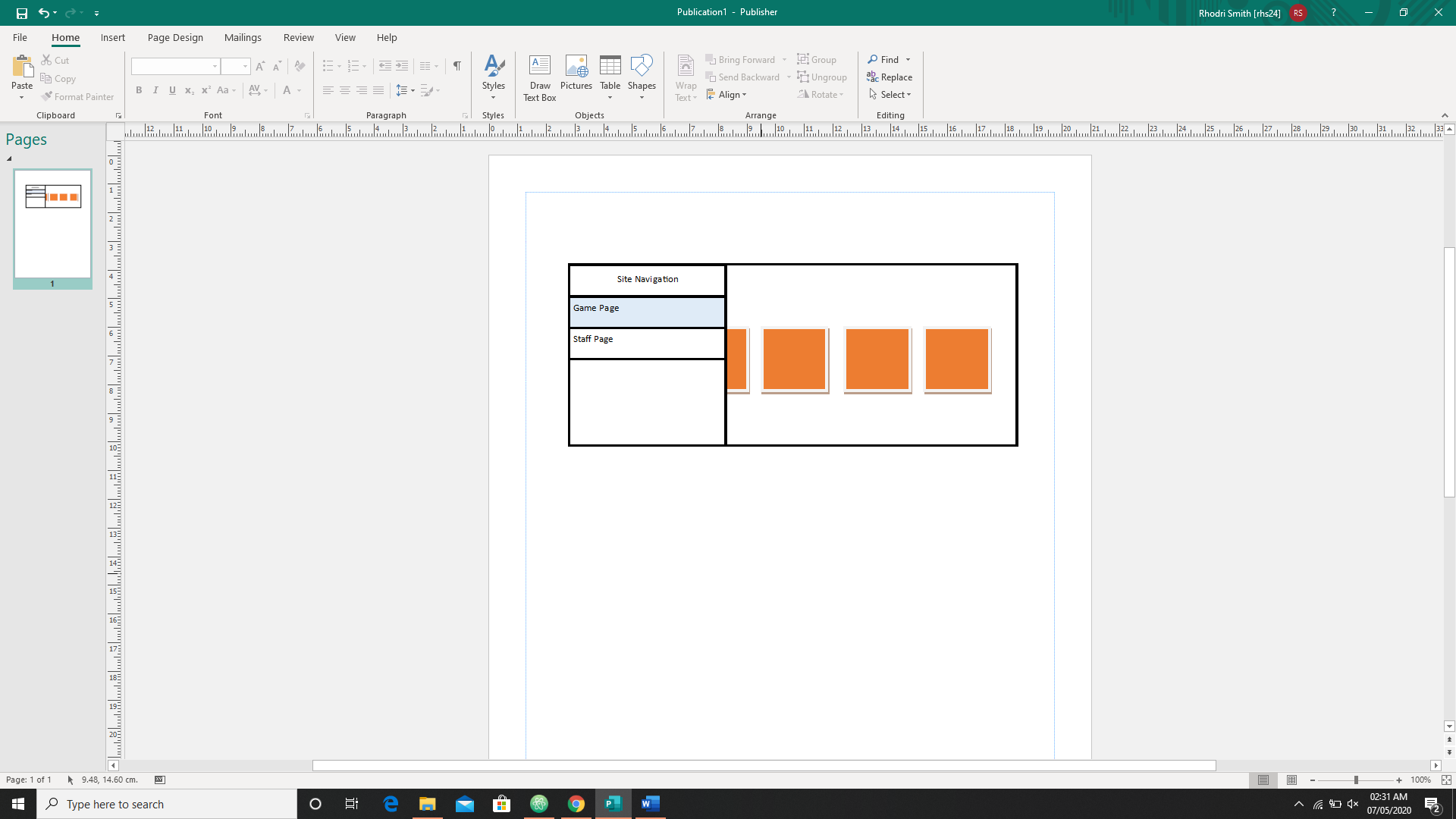


Figure 1 This image is to help the user get an idea of the website if it had a navigation bar

The layout of the page was also taken into consideration. If everything was crammed onto one side of a page, such as a navigation bar, it would look cluttered and unprofessional as seen in Figure 1. We researched Flexbox, a module of CSS that allows a user to easily set out a website through a flexible box by allowing multiple dividers to be set out as desired. Flexbox works by utilising dividers to dynamically set dividers to different sizes in relation to other dividers within the site [15].

### Other sites

In this section we explore some previously developed products that are similar to our specification. We will also discuss the advantages and disadvantages of these products and compare them.

Kahoot [16] is a site similar to the one that is being developed however, one of the downsides to Kahoot is that it does not keep a record of its user’s full data on the site, making it difficult to compare students’ progress. While Kahoot does store the data per game, but if a school wanted to use Kahoot , the school would have to compare the score of every induvial game that a student has played to see if there has been an improvement in the student’s progress.

Also within Kahoot there is no one method for a school to compare students over an array of games. This can be attributed to the website not requiring player to sign up. When the teacher sets a game, the user clicks the link, sets the username themselves (or is set automatically if the teacher has configured it to) making it difficult to track players. The user carries on answering questions until the quiz is completed. The website is well designed and makes taking part in the games enjoyable. Users can also choose to make their own quiz by setting their own question and answer and as a result the site can be used for almost any quiz-based activity. The game host (teachers) need to login but the students do not.



Figure 2: Image of the Topmarks website

Topmarks [17] is another site that we reviewed for the purpose of research and development of the site. This site offers a range of games for 4 different year groups, 3-5 years, 5-7 years, 7-11 years, and 11-14 years. Each year group have their own types of games that a user can play. Topmarks is also split into two sections, one being for maths questions and the other for English questions. Topmarks also has a section that a user can use to choose what type of game they would want to play. In addition, Topmarks also has different types of resources, e.g. the site also has whiteboard resources. The whiteboard resources allow a school to use the website on a whiteboard, allowing groups to take part in the games. Topmarks also offers a mobile/tablet application so that a user would not need to play the game from a computer. The app that they offer is only includes math questions however they have stated they are expanding this provision. Figure 2 is an image of the Topmarks website.

The Topmarks website however does not have users. This means that there is no automatic way to track a student’s progress. The mobile and tablet application does allow for scores to be stored, however.

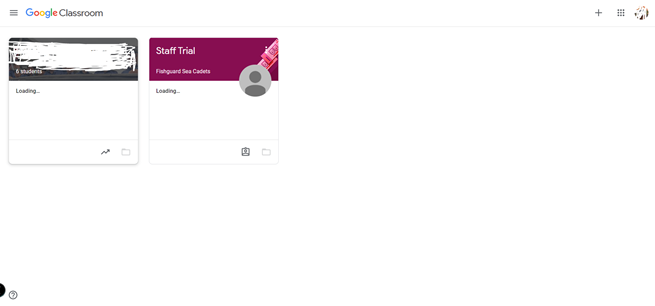


Figure 3: Image of Google Classrooms

Google Classrooms [18] are another way for a school to store tasks online, this site is not directly aimed towards games, however it allows for users to upload tasks of their choice e.g. PDF document with tasks on it, or word documents full of questions that a user can fill in and re upload. The website also allows students to upload their own work that they have been given, e.g. if the student is assigned a task that they need to upload, they upload the task to the website for submission. Once a student uploads a piece of work the teachers receive an email informing of the upload which they are able to download from the website. Classrooms requires every user to have an account and the teacher can keep track with whether a student has or has not uploaded work to the drive. Figure 3 is an image of the google classroom website.

Google Classrooms can also be used to upload information about a class, for example, if someone missed a session due to illness, a file could be uploaded regarding what was discussed within the class, allowing the student to be able to catch up with work remotely. Google Classrooms allows students to be split up into separate classes, allowing a teacher to easily keep track of what is happening within their class, and ensure that work is given to the correct students. Teachers can upload videos to classroom as well as link YouTube videos to it. The Google Classrooms website allows the user to schedule when work is going to become available to the student, allowing the teacher to upload work at the start of the day, so that the students may not see it till the end of the day.

Google classrooms is an easy to use site that has a professional look and feel to it. It is a free website that anyone can use and works on most devices, meaning in young people can also use it from home.

## Analysis

### Aims and objectives

The Aims of the project are:

* Produce a game that a young person can play and understand.
* To improve young people’s mathematical skills while teacher can view their students’ progress through these games and monitor how they are progressing. Through seeing how the student is progressing through the games, the teacher will be able to change the difficulty to those games to move the student forward
* By seeing how a student is getting on within these games, a teacher can evaluate whether a student might be struggling within the class as well as if the student is doing well with their learning.
* To Test the application on the target demographic. Without testing we do not know whether the application is up to standard and is useful in a teaching environment. Teachers will be able to give their opinion on the staff and game site, but the student will only give their opinion on the game section of the website

### Issues and Approach

#### Languages

During the research, numerous issues came to light. One issue was that PHP does not work especially well with JavaScript, passing data from JavaScript variables to PHP is challenging. This made it challenge when it we required to pass data from the JavaScript variables, such as score, to the PHP code. To combat this issue, HTML forms were used to send the details via post to the PHP code. This then allowed any information, such as variable data, to be sent to the database. One of the problems was passing user score to the database, this was an issue as the users score must be sent to the database for a teacher to view their students score.

#### Multi-Page Site

Another issue that arose was how would the website remember the user’s details when transitioning between pages. During our research into the implementation, we considered two methods of retaining user details when transitioning pages: cookies and sessions. Cookies stores information locally while sessions store the information on the server side. The advantage to cookies is that they can store data for long term use, whereas sessions usually only last for a short period of time, e.g. 30 minutes [19]. For the project, sessions were adopted as we did not require data to be stored long-term. Data passed between pages was just the username and table name (school). If the user was an admin, another session was used to state whether they were an admin or not.

#### The Game

When developing games for Primary Challenge, research was carried out to discover the approach for the game interface. Within HTML, a non-canvas approach was taken, as it allowed the webpage to be able to change size based on monitor resolution automatically. This approach would also make the canvas significantly more reusable and giving a dynamic visual response by allowing more graphically enhanced items to be created. However, the primary issue with a canvas approach is that canvas is not designed for games or dynamic changes. Another problem is for canvas to handle clicks, the exact coordinates must be implemented into code, making it more complex. As a result, this meant that a non-canvas approach felt a lot more appropriate for the needed game and design.

#### The Table

We utilised DataTables [6] within Primary Challenge as it allowed for interactivity and advance features while being easy for the user to use. DataTables uses jQuery and JavaScript which allows events such as onclick events for when the user clicks on specific rows or columns. DataTables helps Primary Challenge to achieve a professional look while also providing advance feature such as, a user can select how many items they can see at once on the table. This approach allowed a much more professional design to the website while being still simple to use.

#### Security

Security is a crucial part of web development. This required security precautions being carried out to ensure that our users were safe. One of these security measures that we carried out was hashing of passwords. This stops the website storing passwords in plain text, instead it is just a bunch of random text, so if the database gets breached it stops the hacker having a list of users and passwords. This protects users who reuse passwords. It also protects the users by stopping the admin from viewing the passwords and trying to attack other websites with those login details, since global admins can view the passwords through the database.

The Database will also not store any of the student’s data, other than their name, year, and school. As only these details need to be stored, if the database gets breached, a very low amount of personal data can be stolen.

#### Add New Students

Primary Challenge does not allow new students to be created through the website itself. Schools would contact us to add students and assign members of staff. This also is to restrict the information that staff can access on the students. Only admins with access to the database are able to add users to the site.

A school would need to register their students and staff members when signing up to the website and then need to contact the company to be able to access the database and add a student though the database. This was partly as a result of the ease of adding someone through the phpMyAdmin [13]. This approach reduced the risk of security holes and errors within the website such as duplicates. Users cannot change any data without the support of an admin; this is to lower the risk of anyone gaining access and modifying data they without the correct profligate. For example, if a staff member could change data and accidentally left a computer logged in, someone could change other users’ details. Since a school will only be given one or two admin accounts, this lowers the risk of someone else being able to change details when they should not.

#### Login Page

The login page requires three text-fields: school name, username, and password. Users cannot access the main website without being authorized from the login page as part of access control. However, within Primary Challenge there are four text-fields due to the admin account requiring to be authorized with a separate database which is identified by the first box. For the user however, the first box is to assign the student to the correct school. The user will state which school they are part of by entering their school name.

We did consider a few approaches. The first approach was to use a submit button on the form that caused the webpage to refresh or to redirect the user to another page to sign up.

Another approach we considered was to have a pop box appear for when the website detected a admin. As this was not part of the same form, the code that displayed the pop up would run but then be removed from display as the user is redirected to another page.

We did investigate not assigning an admin to a school, however this required certain users to have multiple user accounts, which is not ideal. As a result, for this phase of development a fourth textbox was used for admin.

#### Admins of Primary Challenge

There are two types of admin within the webpage, one being a school admin, the other global admin. A school admin would be a single account given to the school, which allows the schools to resolve issues without contacting the business. The second admin, global admin, allows for access to all schools and all data within the database. This account is for security reasons and will not be issued to schools. This account like a school admin can resolve problems on the behalf of school such as major errors. The school admin will login the same way as a student account, however the global admin will use the fourth textbox.

## Process

In this section we will discuss the process of producing Primary Challenge and the methodology considered and the methodology that we chose was a variation of scrum. We will also describe the tools that were used to build Primary Challenge in this section.

#### Methodology

For our project we considered the scrum and waterfall methodology. The scrum methodology works by having sprints, which splits the project up into different sections. Sprints usually last for a minimum of a week, and at the start of each sprint, and each day, there will be a meeting between all the members that are going to be working on the project [20] [21]. This methodology was chosen for the project as it allows the project to gain structure and deadlines. The methodology also allows a project to be flexible, e.g. if a section cannot be complete or needs to be changed, the plan can be changed for it. By also having the first day of every week being a brainstorm and research day, it allowed for new ideas and new ways of completing a task to be brought up. Scrum also allows for a quicker way of completing projects compared to most other types of methodologies

We also considered waterfall as another methodology for the project. Waterfall is an easy to follow methodology. It works well when a project is not going to change in any way, however, it was not suitable since the project could have needed to be changed in certain areas or needed to be done quicker. The waterfall also does not allow room for brainstorming and every bit of research would be needed to be done at the start of the project and outlined in as much detail as possible. This meant that we found the waterfall methodology was not a suitable methodology for developing this project.

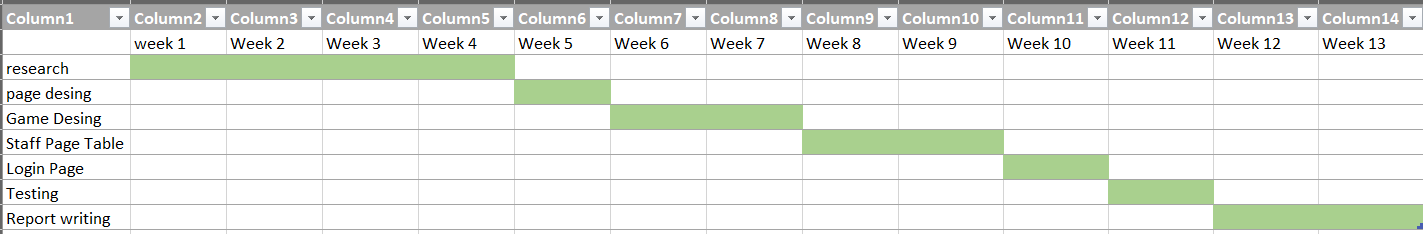


Figure 4: Original Sprint deadlines

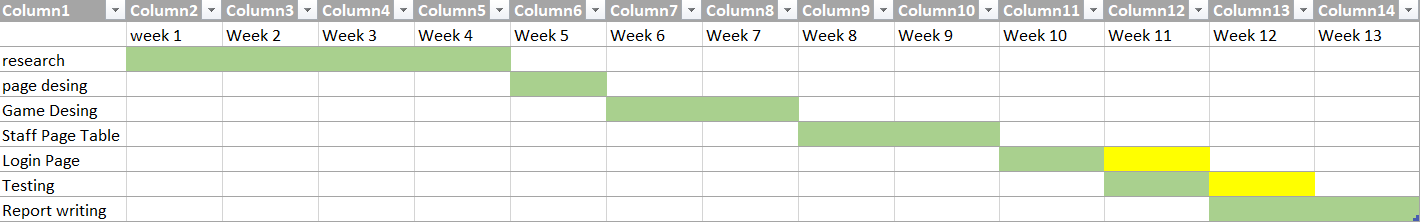


Figure 5: The sprint deadlines including overflow of work onto the next weeks

At the beginning of the project, we split the project into sprints, that can be seen in Figure 4. Sprints were split into a weekly basis to allow all the projects sections to not only fit into the deadline, but to also allow for a few days flexibility in case a stage took longer to complete than expected. The process was followed by every Monday, research was focused on how to complete the specific section that was needed to be finished by the end of that week and in an efficient way. The methodology meant that for most cases, each section had from a week to two weeks to be completed. With this in mind, the second week still included a research section. The second Mondays were used to not only look into different ways of fixing or improving the code that had been completed and on how the next week’s work would be mapped out. For example, Figure 5 shows when during the login page, an issue rose when the hashing function stopped the code from running, resulting in the sprint running over into the testing week. This pushed testing into the week set aside for writing the report. The sections in yellow demonstrates where work overran its intended deadline

However, Scrum had its limitations. For scrum to work effectively, teams should consist of more than one person and the team should have a meeting at the start of the sprints. Scrum should also have a scrum master. The meetings would outline the criteria for the sprints and team members will discuss how they intend to approach the sprint. As it is not possible to have a meeting with only a single individual present and working on the project, it was decided that this would be replaced with a brainstorming session to consider different methods of completing the sprints. At the end of each sprint, the program went through testing to ensure it worked as intended.

A Project Goals document was also created, this outlined sections of the application that will needed to be completed. The document also states how the developer wanted the project to be completed, e.g. how the database was meant to be set out. The document also has some basic use case diagrams that show how the user was meant to get around the website. The Project Goals document can be found in screenshots included in the appendices, figures 49 to 53.

Figure 41 and Figure 42 in the appendices shows the diary that was made at the end of each week within the project. The diary gives a brief description of what was done for each week of the project. This was originally created in separate documents however have been moved into one document.

#### Software and tools

Software and tools are an integral part of software development as it allows the programmer to not only write the software’s code, but also do more advance task such as syntax highlighting version control. Without version control and backups, projects can run into problems when parts of the project could just stop working or even complete loss of the project if a computer failed. Companies cannot afford data loss to occur, as it costs time and money. This is where version control and backups can help a company. Using version control, if a section of code breaks when development is happening to it and a fix cannot be found, a user can just remove the corrupted file with a known good file. However, if the project needs to roll back to a previous state, all changes made have to be discarded.

For software control we used Git. Git is an open source system that allows version control for projects of any size. Git is used through the terminal window and requires the use of commands. These commands can be found online through Gits own website, or through typing in the help command in the console. You can create a repository on a computer that allows a user to push and pull and push files to a remote server. A user can also have git repositories from more than one computer and users, allowing the user to work and upload from almost anywhere. Git does require a user to have access to the internet to push and pull documents.

As Git supports an external sever for offsite backups we used GitHub [22]. GitHub allows all files including code, config and graphics files to be stored off site. GitHub also allows repositories to be shared with other developers if we wanted to get more developers onto the project.

Atom [23] allows GitHub to be directly integrated into it, allowing the user to create push and pull requests at any time. If changes are made within the code, the application will detect the changes and place the files into an ‘Unstaged’ section for the user review before committing to an external repository. This made it simple to work with Git.

The GitHub repository is stored at <https://github.com/Rhod4/MMP>. Stored here are all the files that where used within the project. These include the website, the designs and the testing documentation

# Design

This section will talk about the design of the program, this includes the overall look of the project, to the code that is within the project. There will also be diagrams that shows the flow of the website, from how the user gets from one page to another. The design will talk about every part of the website and in certain sections will go into detail about the functions.

## Visual look at the website

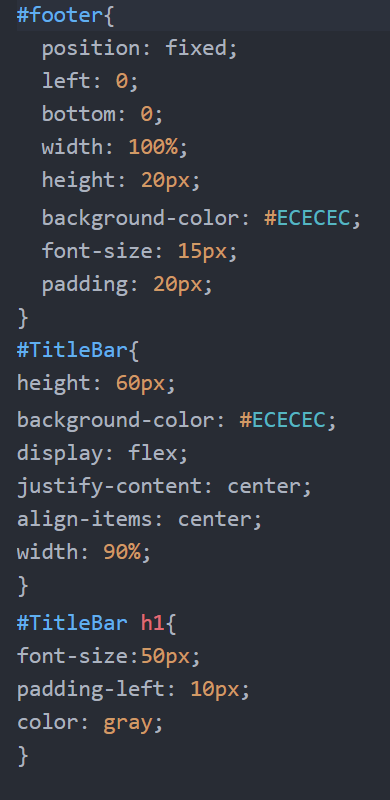


Figure 6: Main CSS file for the site

#### Multi Page Visuals

Since the pages were going to have the top bar and the footer to be the same through the site, it was decided to use one CSS file to store this data. This gets rid of duplicate code and would allow the developer to easily change the look of those sections from one file, making maintenance on the website easier. Figure 6 shows the main CSS file for the website.

Flexbox [15] was also used within the site as it allows the pages to be set out in a professional manner and allows easier control of the pages, e.g. setting out the dividers to all be in the centre of the screen. Since flexbox is built into the CSS framework, it is incredibly easy to use and maintain.

#### Look of the website

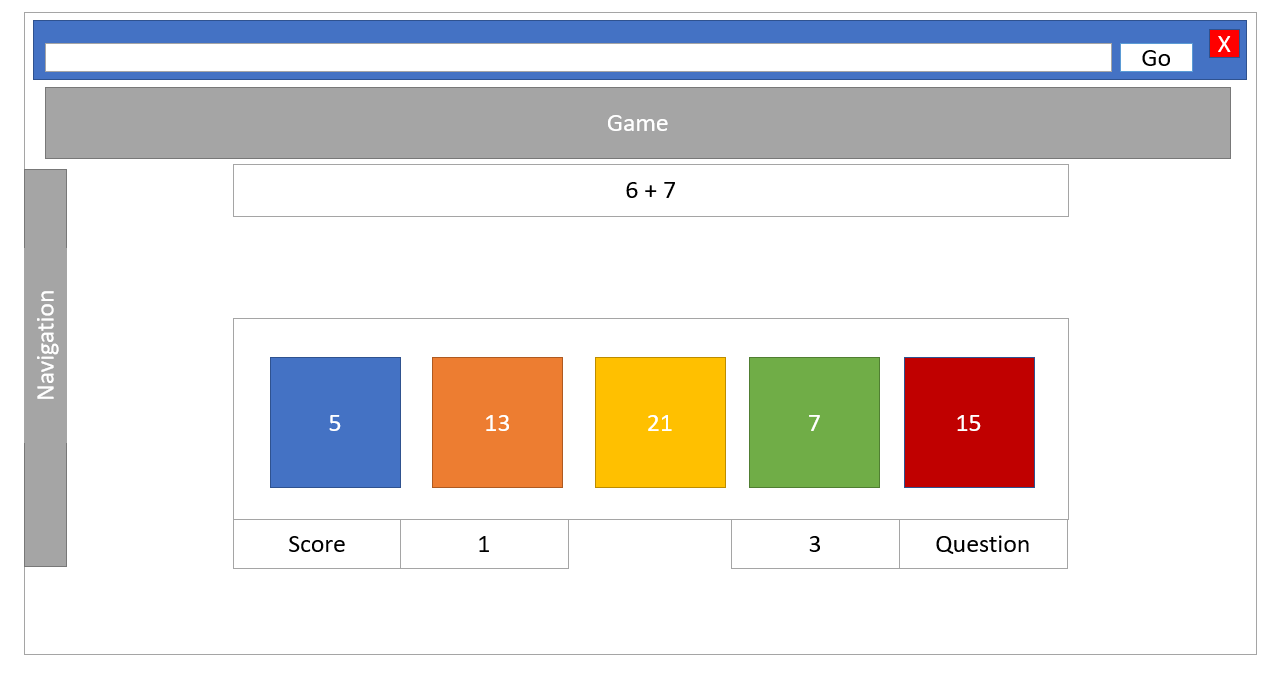


Figure 7: Navigation bar on website

There were a few designs were considered for the website and two were developed into a prototype for further testing. The first prototype was for navigation, which allows a user to navigate through the website to the desired pages. The navigation bar would be set to the left-hand side of the screen and would pop up when the user’s curser hovers over. There would have been a small side bar sticking out on the left side of the website, which would contain the text Navigation, as shown in figure 7. The navigation bar would allow a staff members and admins entry to the game and staff pages, while the students will only have access to the game page. This design also incorporated the way to edit users by a pop-up box showing in the centre of the screen. In order to edit a student the staff member will need to click the ‘edit button’ and then to remove the user they would click the ‘remove button’, both are on the table in columns. This prototype would have incorporated a text box for a staff member or admin to add difficulty

The second design for the website is the one that is closest to the implementation. This design takes the user straight to their pages, e.g. staff go to staff page and students go to the game page. This design means a user does not have to navigate the site manually. The site uses the same method as the other prototype, of a pop-up box to allow the staff member or admin to edit a student. In this design the edit buttons are not in the table, instead the user needs to enter the pop-up box by tapping on a user’s row, e.g. their name, where they can also edit or delete a user. This design also incorporates a multi choice option for the difficulty.

The design that was used within the project was created into a prototype and the screenshots for it can be located in the appendices, figures 43 to 48.

#### Layout

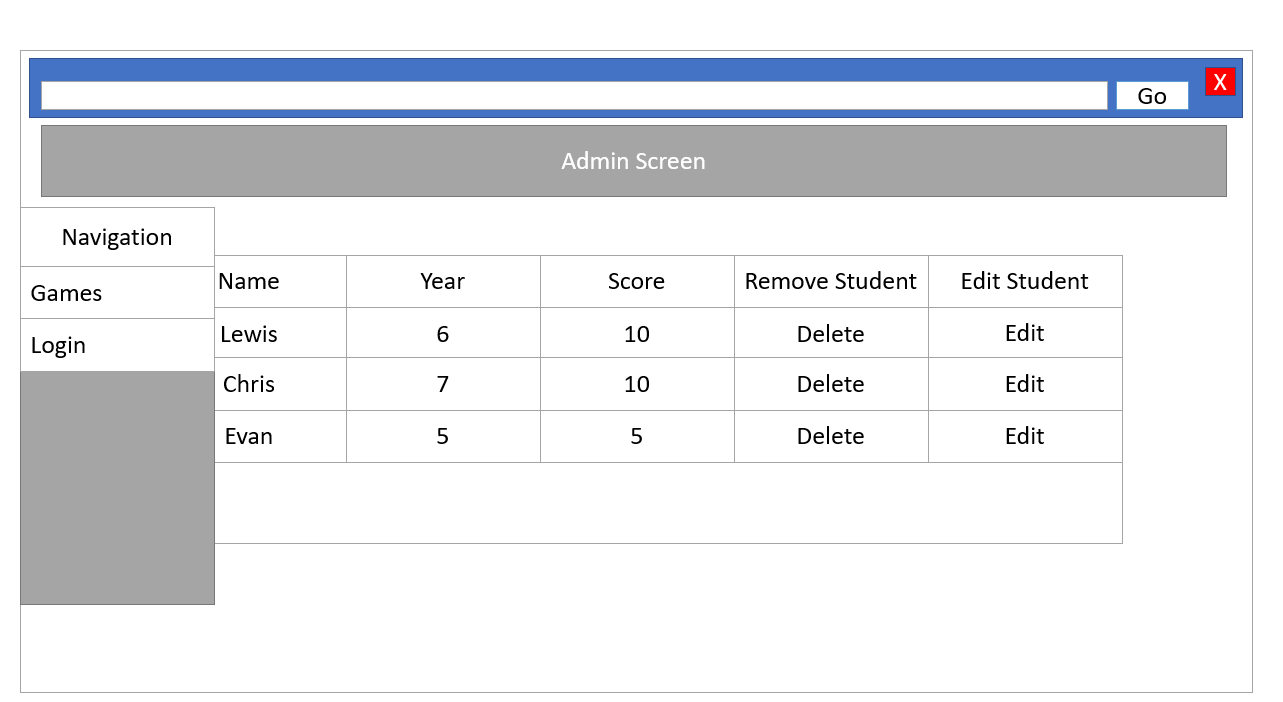


Figure 8: Side bar navigation

The layout for the website was based on the placing of everything to be in the centre of the screen, nothing on the left or right side, resulting in the website not requiring a navigation by the user. During our prototyping we did investigate using a side navigation and top navigation as seen in Figure 8, however, we felt that by having everything in the centre of the screen it makes the website look more professional. It also allowed the website to change size based on the monitor resolution

### UML and other diagrams

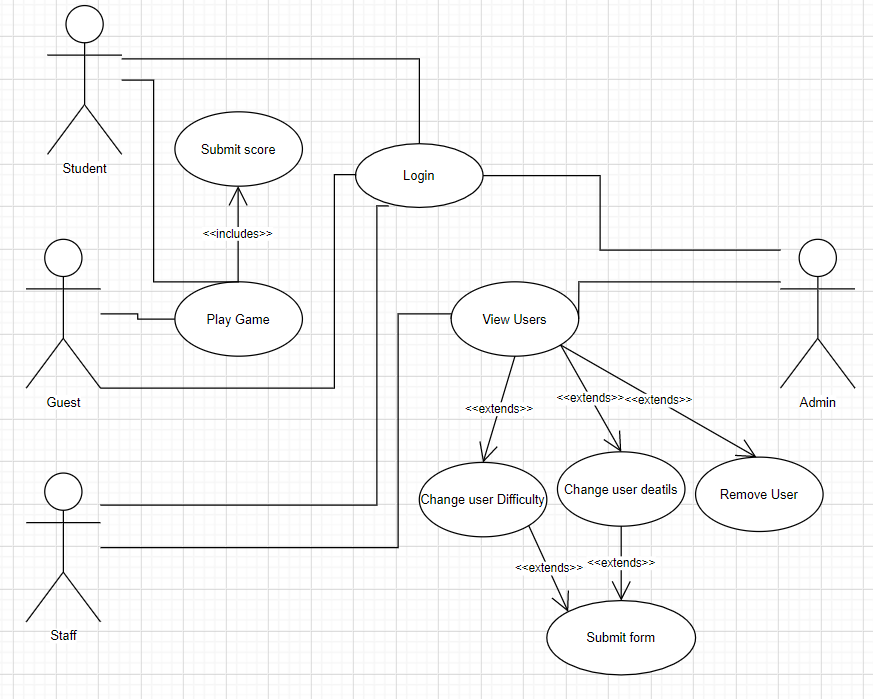


Figure 9: Use case diagram for all users

Figure 9 displays a user case diagram for the system. This diagram shows all the users that will use the website. Starting from the top left, the first actor (stick figure), shows the ‘Student’, which as shown by the diagram, can enter two states. The first being the ‘Login’ use case, which on the website will be the login screen, and the user will login to their account here. The next use case they have is for the ‘Play Game’ use case, which is the game page on the website. The ‘Play Game’ also has an ‘include’ next to it, this is the ‘Submit score’ use case, this state may not need to be run, e.g. if the user is a guest their score is not sent to the database.

The next actor is the ‘Guest’ actor, this actor connects to the ‘Login’ use case, but they do not login, they will just click the guest button. The ‘Guest’ also links to the ‘Play Game’ use case as they also can play the game, but their score does not get saved to the database. The Submit form will not fully run for a guest user, as nothing will happen within the PHP code since no details have been set.

The third actor on the list is the ‘Staff’ actor, this individual links to the login use case as they also need to login to use the site. The actor can also go to the ‘View User’ details with this site which is the staff page on the website. The staff member can also change certain details of the users, e.g. the level of difficulty. This means that they can go down to the change user difficulty section, but they do not have access to the other extends.

The ‘Admin’ is the last actor in figure 9, they need to login to access the system and as the ‘Staff’ actor, they can also go down the ‘View User’ use case and they are able to access all the extended sections for the use case, e.g. ‘Change difficulty’, ‘Change user details’ and ‘Remove user’

We developed User flow diagrams to help us to design the website. In this section we will describe the UML diagrams we developed.

User flow diagrams are used to show the flow that a user would take through the website. They help give a user an easier to follow diagram that explains what is going on while going through the website.

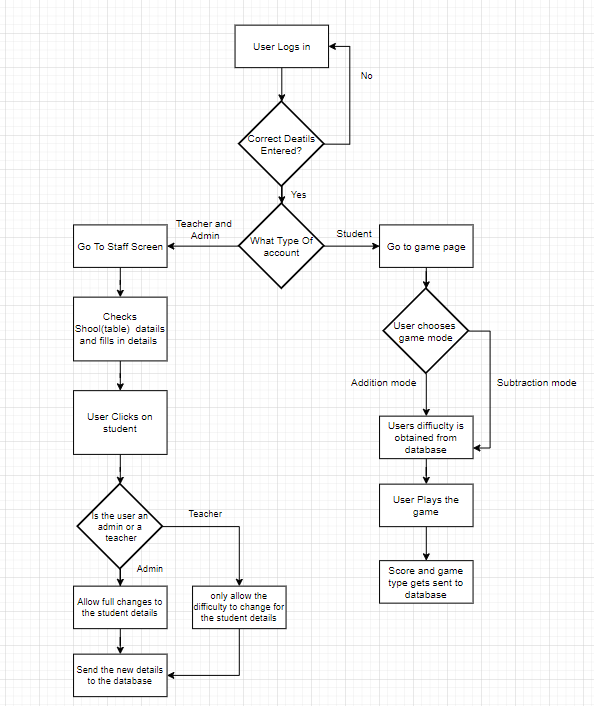


Figure 10: User flow diagram that is for all users

In Figure 10 we can see the basic flow for the website of the front end and backend interaction. Once the user has been authorised to access the website, the website then checks to see the type of website that was used.

If the user is an admin or staff member, the diagram would go down the left branch. The left side shows the user being taken to the staff screen. The system then checks the detail of that user, e.g. the school name to fill in the table with appropriate details, e.g. students. The next cell on figure 10 happens when a user clicks on one of the table rows, this brings up a box for the user to change the students details.

If the user is a staff account, they can only change the level of difficulty of the questions for the student. But if they are an admin account, the admin can change all the details that are shown on the table. The two split sections then reconnect to the ‘Send the new details to the database’ cell. This cell would activate when the user has clicked the submit button, sending the new details to the database to change update the previous details.

The ‘Go to Game Page’ cell shows where the user would be taken if they were the student or guest profiles. The ‘User chooses game mode’ refers to the game page of the website and the page will contain the game that the user will be playing. The user chooses the game mode that they would like to play the game on, this is a diamond as there are multiple options that change how the game can run. However still all lead to the same outcome for the rest of the game. The ‘Users difficulty is obtained from ‘database’ cell, which accesses the level of difficulty from the online database. The player does not have the ability to set the difficulty this is obtained from the database. The ‘User Plays the game’ cell would be for the user when playing the game, e.g. answering the questions. The users score will change based on the amount of correct answers they score e.g. a correct answer gives +1 to their score. At the beginning of the game, the score will be 0. The ‘Score and game type gets sent to database’ cell and shows the users score being sent to the database. The score will be displayed on the staff table as a new score.

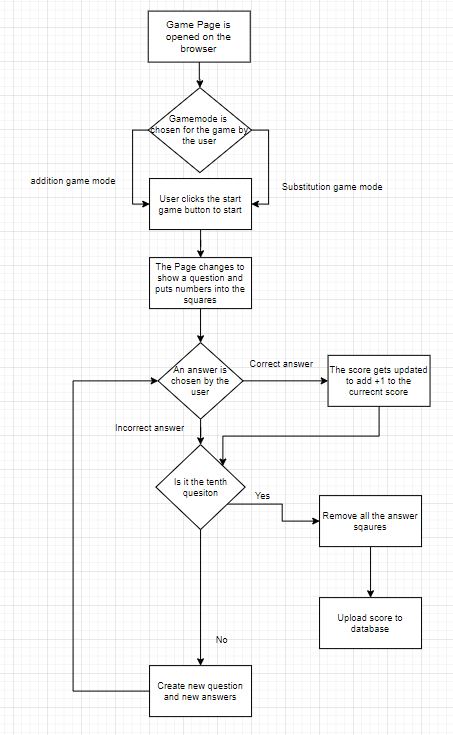


Figure 11: flow diagram of how the game works

Figure 11 shows the flow of the user when they play the game on the site. The ‘Game Page is opened on the browser’ cell is the first state of the application within the diagram, which occurs once a student has logged in from the login screen. The colour squares that the user will click on are generated here. ‘Gamemode is chosen for the game by the user’ cell which states whether the game enters the addition or subtraction game mode. This is selected by the user by a pop-up box.

The ‘User clicks the start game button to start’ cell discusses the actions that occur once the user starts a game. The next cell is the ‘The Page changes to show a question and puts numbers into the squares’ and states how questions and answers are handled. In this section the answer is displayed in the text in the different colour squares and the questions replace the start button. The diamond cell stated as ‘An answer is chosen by the user’ calculates whether the user clicked the correct answer. If the answer is correct it will go down the correct answer branch and if it is incorrect it will go down the incorrect branch. Both the branches go back to the cell ‘is it the tenth question’ eventually but the correct answer branch goes on to cell ‘The score gets updated to add +1 to the current score’ first. In the cell ‘The score gets updated to add +1 to the current score’, the users current score is increased by +1. The game will then check which question the player is on at the ‘is it the tenth question’ cell. If the user is on question 10 (the length of the quiz), the answer area gets removed and a score is uploaded, the section coming off of the ‘is it the tenth question’ cell with the label ‘Yes’. If the user is not on question ten, a new question is generated, until question 10 is reached.

## Detailed Design

### Login page

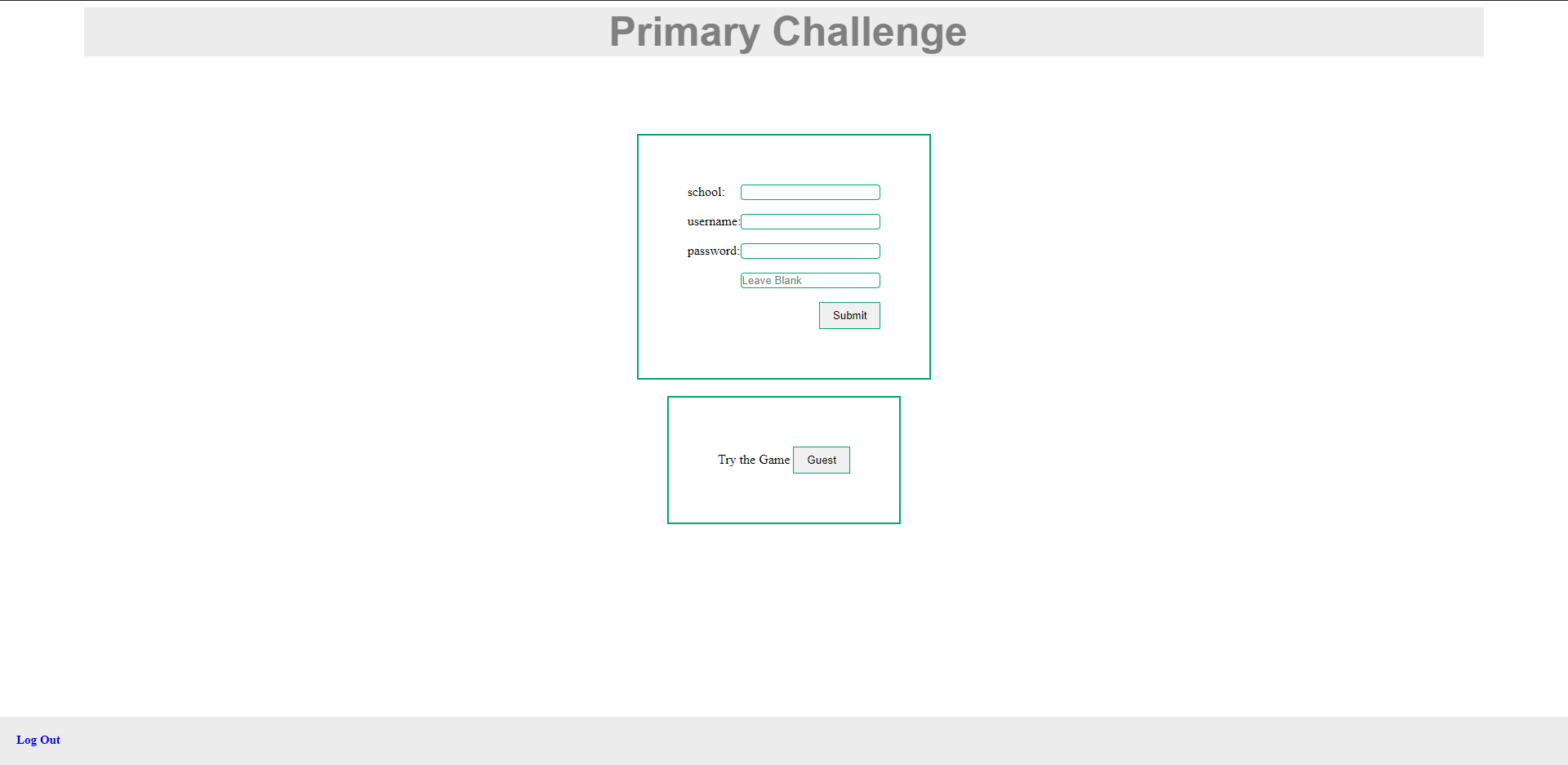


Figure 12: Image of login Page

The login page’s design is designed to look user friendly, allowing people to easily go onto the site without any issues. The user needs to enter their details in and click submit, they will then be taken to their appropriate page. The user must enter their school, username, and password. The fourth box would only need to be filled in by an admin and would be the school name that they wish to connect too, which is why it states to leave blank. Figure 12 is the image of the login page that the user views

The page uses HTML forms and PHP to complete the login and to take the user to the correct page, the game page for the student or guest. The pages that the user will be sent to will be dependent on the type of user; a student and guest will be taken to the game page, while the admin and the staff accounts will be taken to the staff page.

When the user clicks the login button, the systems sends the details to the database to check through and return the matching user accounts to the system. The username is checked and if it matches any usernames within the database, the user password is then verified with the password\_verify($password,$user['password']) function.

Within the project the password verifier is used within an if statement. If this returns true, the login code continues, otherwise it terminates. After the verification, the system works out what account type the user is, for example an admin or student. If the user is an admin, the system then checks what type of admin they are, e.g. school or global admin, and assigns the admin their set of schools based on details they have filled in.

If it is a global admin, the school session gets set to the text that the user has entered into the fourth textbox in the login form. If the user is a school admin, the school session is set to text that is in the first text box (school box). The user is then taken to the staff page, where they will have more access than a teacher. If the user is not a global admin, the fourth box is ignored.

If the user is a teacher, a school session is set to the school they entered in the first input box, and the admin is set to be null as they do not have admin privileges. The user is then navigated to the staff page automatically.

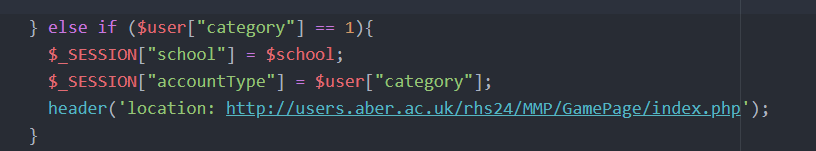


Figure 13: Student account login statement

The students will login the same way as the staff members, and the code runs in the same method, being split up through an ‘if’ statement depending on their type. The system also sets sessions for the school and the username they entered. The student will be taken to the game page, not the staff page. Figure 13 shows the statement that will run if the user is a category one user (Student);

The users types that are stored on the database’s category column, e.g. staff member, student or admin, is all stored as integers, 2 being for admin accounts. 0 being for the teachers and 1 being for the students. User roles are stored in this way to save server storage to resolve possible errors, e.g. spelling errors.



Figure 14: Guest button on login screen

The login page also has a button under the login form, called the guest button. This is the button that a user would use to test the site and to be able use the system without logging in. The intended case for this is to allow a potential user to see whether they want to use the system or not. The guest section works by setting just two sessions, the username and school both to the string of guest. The user then gets sent to the game page where they can test the game. They cannot access the staff page however, since it would be empty for them anyway. Figure 14 shows the guest button on the login page.

### Game Page

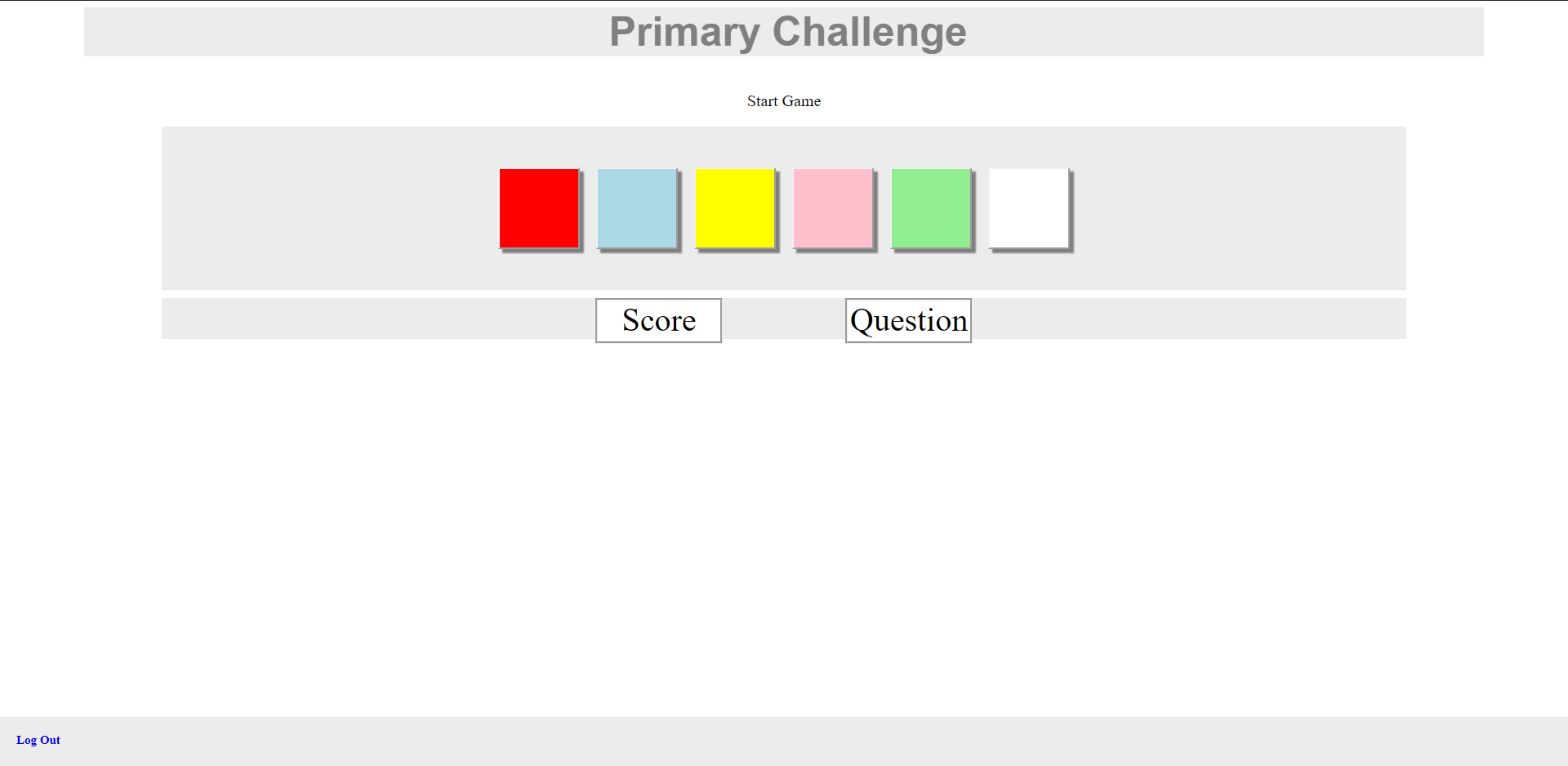


Figure 15: The game page

The Game screen was designed to be easy to use. The game page works by requiring the user to click on a ‘click me’ button, which allows the user to have time to load the page before the questions begin, e.g. if the game starts to use canvases, slower connections could become an issue if time was limited. Once pressed the user clicks on one of the coloured boxes to answer a question and the score is totalled up based on the amount of answers the got correct. The game page is set in the centre of the screen, the question being in the top centre of the page and possible answers below the question, making it easier for the user to recognise the correct box to click on to give their answer. The boxes include text are reasonably sized for the user to identify the answers. Figure 15 shows the game page that the user will play the game on.

JavaScript is used within the page to add interactivity. By creating the squares in JavaScript, the developer can easily change how many squares appear on the screen at once, such as having more or fewer squares in the page.

The users score is sent to the database through a form once they have completed the game. However, this form is not is not visible to the user as it is hidden. The form is filled in and submitted automatically, by the JavaScript. If this was a manual procedure, someone could lie about their score or could just forget to click the submit button to save their score.

The game works by giving the player a random question and not stored questions. This lowers the chance of a user getting a repeat question and does not require a user to put every possible question they want into a database.

The question algorithm works by creating two random numbers, the two random numbers are then shown to the user with the game mode’s type between, e.g. 3 (first random number) + (game mode) 2 (second random number). The two numbers are then processed to create an answer. Which is then used within the application as an answer to the question.

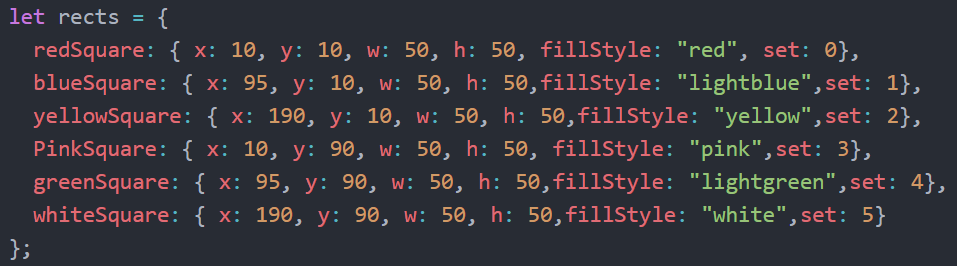


Figure 16: Array for the colored button's details to be stored

The game is based on users selecting an answer that is printed onto 6 coloured squares. These squares are not coded onto the HTML itself; they are instead inserted into the HTML by JavaScript once the page has loaded. The array stores the coordinates for the squares, the hight, width and colour of the squares. This allows the developer to add more or remove the number of squares when needed. Adding and removing squares would be used with the difficulty scale that the teacher sets, however this is something we consider for future works. Figure 16 shows the array that holds the details for the squares



Figure 17: DivMaker function

Since the squares are not created within the HTML, the JavaScript must create them instead, function DivMaker() was called. DivMaker() is a function that loops over the array ‘rects’ and places them in the correct space depending on the values of the objects within the array with the array being seen in figure 16. The DivMaker() function can be seen in figure 17. The squares are drawn into the ‘JavaGame’ element as new dividers, placing them in the correct place on the web page.

A ‘switch’ statement is utilised by the program to work out what mode that the gameMode variable should be assigned. The statement calculates the difficulty that the user. Difficulty is controlled by significant figures. for example for a difficulty of 1 the player would be asked questions to one significant figure (for example 5 + 9) and a difficulty of 2 the player will be asked a question two significant figure(for example 10+45). This switch is also used to control the end of the game by utilising an exit function which cleans the screen and displays the user score.

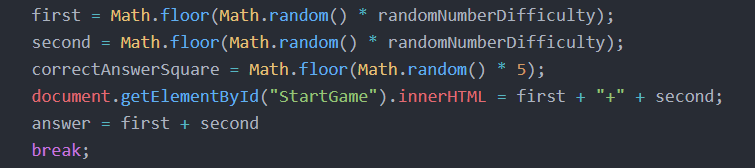


Figure 18: Random Question Generator for addition

To generate the questions, a random number generator was required. We utilised the built-in number generator in JavaScript, Math.random [24] to produce the numbers. However, Math.Random() produces floats between 0 and 1 [24] and to compensate we used the following algorithm:

randomNumber=Math.Floor(Math.Random × randomNumberOfDifficulty)

Where ‘Math.Random()’ returns a random integer, and ‘Math.Floor()’ function rounds that integer to the closest hole number and then returns the integer. Before the ‘randomNumber’ is chosen ‘randomNumberOfDifficulty’ is multiplied to the number. ‘randomNumberOfDifficulty’ is the difficulty for that user. Its multiples the Math.Random()’s number to create a larger number, e.g. after the function fully runs, if the difficulty was set to two, this means ‘randomNumberOfDifficulty’ would be set too 100. This results in a figure that can go into double digits. Figure 18 shows the random question generating code for the project.

This function runs twice for the question, the first time for the first question and the second time it runs for the second question. This algorithm was utilised to generate incorrect answers within the project.

If two squares in the game ended up having two of the same answers, and one was the correct answer. If they did share the same value, the incorrect answer would be remade, and the correct answer would be retained. This was done by using a for each loop that ran for the number of items in the array that stores the rectangles,

Currently within the game there is an addition and subtraction option that can be set for gameplay. The game is currently limited to 2 significant figures at the moment. However future additions for this work would be to implement a divide and multiply feature.

The add or subtract questions are created by using the math.random function and the math.floor function. If the random number that is generated is higher than 0.5 it rounds the number up, otherwise its rounded down. If the game mod is set to subtract, we ensure that all answers are smaller than the question and larger for addition game mode.

When the user presses a square the AnswerChecker is called. AnswerChecker checks to see whether the selected answer was correct. It checks whether the question was correct by checking the correctAnswerSquare variable that contains the answer. ScoreUpdater is then called which updates the correct score, before calling RandomQuestionGenerator to generate new questions or to exit the game. The user is then informed whether they were correct or not through seeing their score increase or stay the same.

If the user is a guest, no data is saved from the game to the Database as only logged in users have data added to the database. This is because no one would be able to view the guest’s progression, meaning there would be no need to record the progress of the guests

### Staff and Admin Page

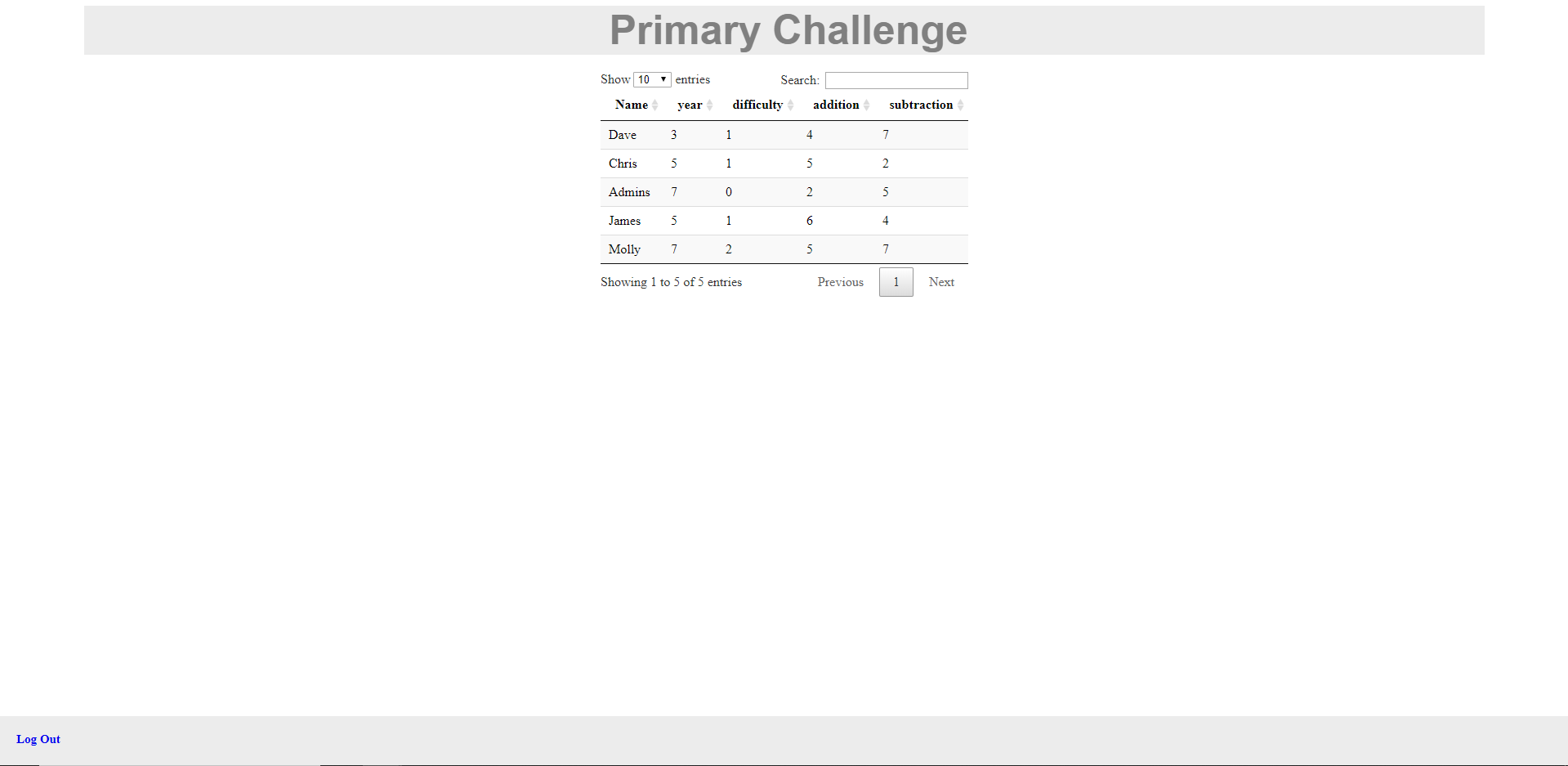


Figure 19: Staff and Admin Page

The staff and admin pages consist of a table which contains students as seen in figure 19. All data relating to the student with the exception of users ID number and password is displayed in this form. The users ID numbers are not displayed to eliminate confusion as this cannot be changed by the user and would not make sense to the user. Staff and admins can edit the students details of students by selecting the edit button on students’ pop up form. When a student’s row is clicked on, a form comes up with details that can be changed. If the user is an admin, they can change all details and add and remove people. If the user is not an admin (such as a teacher) they can only change the level of difficulty of the game for the student. Admins can change all the details of the user within the database other than their number

When details are changed, the details are pushed to the server via a POST. This is then written to the database by PHP on the server side. If the user selected “remove” a hard delete is carried out where the individual is deleted entirely from the database, which in turn removes them from the table.

The table is generated on the server side before being sent to the client. The server uses PHP to query the database for students before looping over every student in the database, echoing the text into the correct columns in the HTML page before transferring to the user. If the user is not an admin, they can only see the students however admin can see everyone on the database

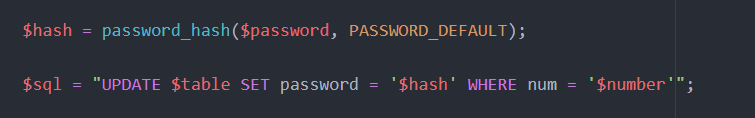


Figure 20: Hashing the passwords

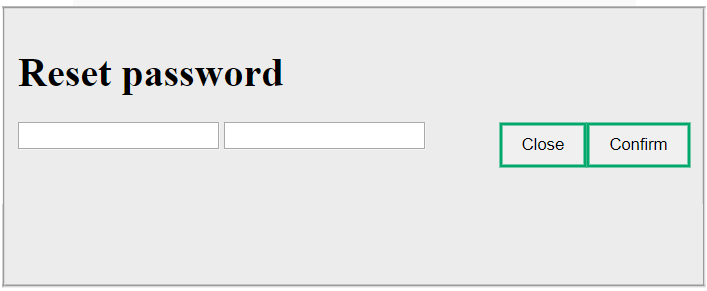


Figure 21: Image of the reset password pop-up box

The admin can also allow users to change their passwords, which is useful if users forget their password. They also give new users passwords when they are created. Passwords are given in plain text which are then hashed before being sent to the database. Figure 21 shows the function that is used to hash the passwords and set the new hashed passwords on the database. Figure 22 shows the reset password section of the webpage.

### Logout function

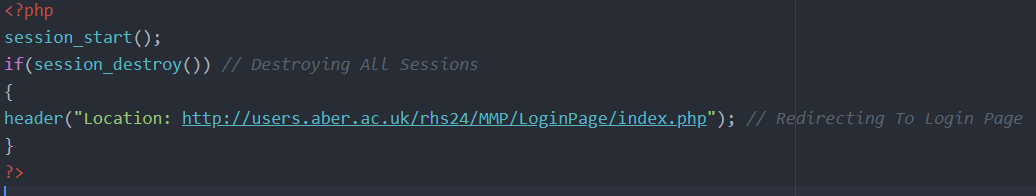


Figure 22: Logout functions code

The logout function is contained within its own file. This function simply destroys all the active session that have been created, before redirecting the user back to the log in page. The session must be destroyed to ensure another individual cannot use an active session of someone else’s account to cause malicious damage. Each page has a logout button that is contained within the footer of the page. This is needed for security within the website as without this, someone could be able to access someone else’s account if the sessions have not ended yet. The code of the logout can be seen in function 22.

### Database connection

The database connection file allows the website to connect to the database. It severs two function, firstly allowing a connection to the database via the OpenCon() function and secondly closes the connection to the database. Within this function four variables are stored: the database host address (db.dcs.aber.ac.uk), database username and password and database ID (‘cs39440\_19\_20\_rhs24’). This is to allow the connection to the ‘cs39440\_19\_20\_rhs24’ database. The code for the database connection is a very basic connection function. The username and password of the database are not shown within this document for obvious reasons.

## Other Relevant Sections

### Security

For each page within the website, the user must be logged with the exception of the login page. The website checks if the user has any sessions set, these session being the username or accountType. If the username is empty, the page will redirect to the login page using the header function. The accountType session is there to stop any other accounts, such as an admin from getting to this page. If they do manage to find where the page is stored, they get redirected to the login page.

# Implementation

In this section we will discuss how we implemented Primary Challenge.

### The Staff Page Table

DataTables [6] is a third party library we used within Primary Challenge to make tables easier for the user to use. Without the use of DataTables, the user experience of Primary Challenge will degrade. For example, if the school had a lot of users, the table may require the teacher or admin to scroll down a vast amount to get to a specific user. DataTables adds functions to HTML Tables such as a search box. Without DataTables, it would require our own implementation which would be difficult due to time restrictions and complexity.

In the early version of the project, an admin would log in to the website and would then be asked what school they wanted to connect to, through a pop-up box. Due to changes within the system, we then had to ask the admin at the login screen to then state which school they logged into. When the user pressed submit, it would automatically post to the server, which refreshes the page. To get around this we placed a extra input box on the log in screen. We did consider navigating the admin to another page at first to ask them what school they wanted to connect to but decided against due to changes in the website design. This would have also required another unnecessary page on the website.

We found that using PHP was more demanding than anticipated due to the way the website was hosted. As the website was hosted on Aberystwyth servers, we needed to connect to the website via the Aberystwyth VPN. As we used windows, to SSH into the server we needed to use Putty [25] and do a ‘fixwebperms’ to set the correct file permissions. Due to closers related to COVID-19 we could not access the server by visiting the university and found this approach very slow.

Another demanding area with PHP was that when we received errors, no errors were displayed originally meaning that we had to use statements such as ‘$conn->connect\_error’ to show the errors. This slowed progress down as it was never clear what section the issue was being created on.

#### Different questions types

We did consider implementing different types of questions in Primary challenge such as spelling questions. For example, a shape would be shown (banana for example) and players would be asked to click the correct spelling from boxes (such as banana, ban\_na, \_anana). We decided not to implement it for a few reasons. Firstly we did not have to implement a question bank for mathematical questions as they were automatically generated, but one would need to be for this. This removed the need to store questions and ensures that players do not get asked a repeat of questions. Another difficulty was that it is difficult to make words in English automatically. While the word Banana can be easily randomised to contain other strings or an underscore. for example, these rules are difficult to implement with a random generator. Machine learning could have been utilised for this; however this was outside the scope of the theses. Another approach could have been to generate random words from an array, which could get its details from the database. However, this array could end up being very large and complex for a user to read.

# Testing

The testing section will talk about the overall approach to the testing, which includes what was tested, how it was tested and who was asked to test the website. The section also talks about the tests the developer completed within the project to make sure everything was tested thoroughly and correctly

## Overall Approach to Testing

Manual testing was carried out to ensure that the website worked as intended at every stage of the project. Each function was tested to ensure that it worked as intended by the developer at the time the function was completed. However, this did not result in usability testing as the development team did not meet the target demographic for the development. We did not do unit testing on the website because it was already planned to do user testing and manual testing. It was chosen not to use it because that there are no built-in unit tests within JavaScript, this means that third party software would have been needed to be implemented and learned. Since two types of tests where already being completed and the developer needing to learn/understand the unit testing software, it was chosen not to use them.

However, Unit testing tools such as Selenium [26] and Mocha [27] were looked into, to see if they could be used within the project. These are two third party unit testing software. They both allow for lightweight and complex tests to be done on a website with relative ease.

## User Testing

The initial plan was to get participants of our target demographic to test the application for real world feedback. We planned to ask participants about the design and functionality of the website. However, due to the global COVID-19 pandemic restricting on movements and the closures of schools, deemed this impossible to carry out. Instead of the target audience testing the website, a different audience had to be used, this group included more computationally adept users who tested the website remotely.

This new group of testers was limited to 7 participants (6 male and 1 female, aged 20 – 60) but because they were not part of the original target player demography they were not able to test the website with admin control for security reasons. This study was carried out at the end of the project, allowing for the testers to test the whole site and not to have to work around unfinished sections. Since a younger demographic had originally been chosen, they would have found it difficult to test an uncompleted website. With this in mind it was decided to user test at the end of the project.

We carried out user testing with participants to test the website, as this would allow for a higher chance of bugs and issues being identified. The users involved in the testing were asked to offer opinions about the site in their test reports on the design, usability and their overall judgement of the pages. This included the login page, the game page and the staff page as well as an overall score and comments.

Participants were given a document, figure 26, that they were asked to fill in and hand back when completed. They were asked not to write their name or to write any comments that could identify them on the document to ensure anonymity. They were able to return the completed documents by any means and the files were then uploaded into a folder on the Aberystwyth file store, since they had no correlation to the user, and the documents could not be linked to a user. The only way to access this data was to get into the developers Aberystwyth University account.

In the appendices there is figure 26, which shows a blank test table that was given to the user. There is also, figure 34, which is another document that was given to the user. This other document is the ‘Usability Study Participation Agreement’, which gives the user information about their input and explains the confidentiality of it.

As a result of the feedback some changes to the website were made. For example, it was pointed out that the yellow used on the website was very sharp on the eye. This was replaced by a gentler shade of jade.

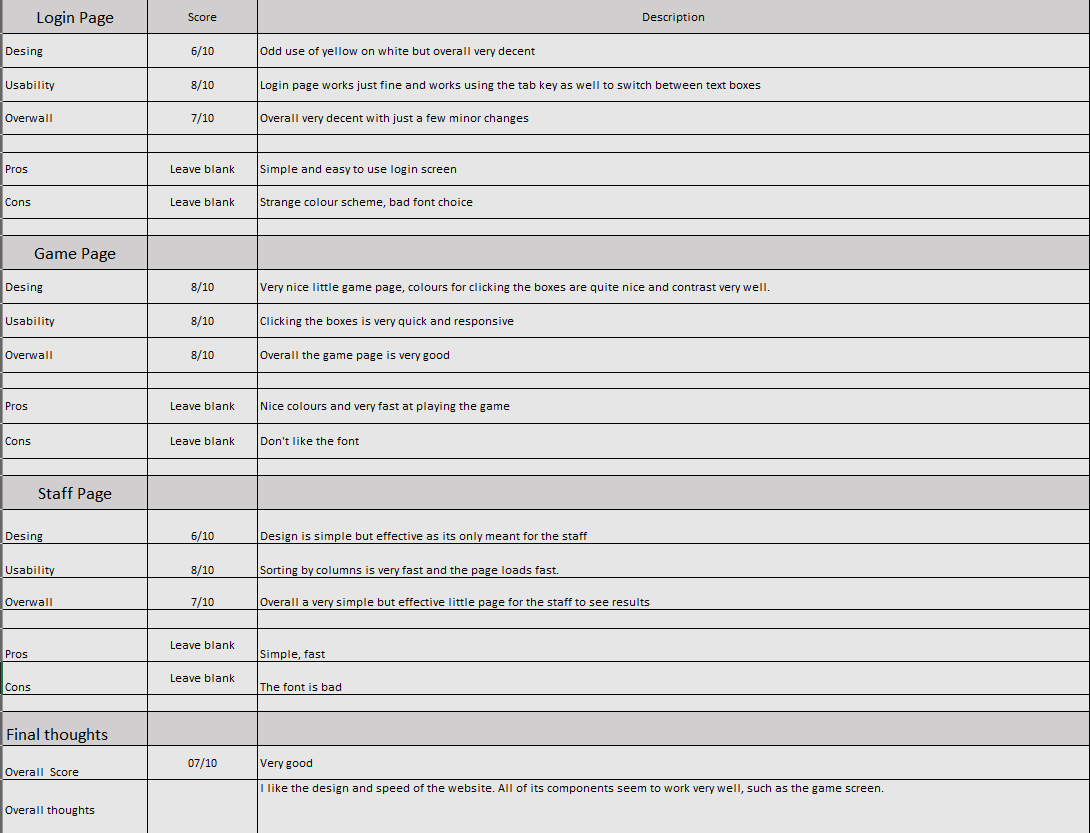


Figure 23: One of the User Testing reports



Figure 24: Second User testing document

In figure 23 and figure 24 the reader can see some of the responses from the testing. The feedback given was then used to considered and some changes where then made to improve the website. The appendices contains all the user test for the website, these are figures 27 to figure 33

## Manual testing

In the appendices, within the tests heading, there are figures that show the manual tests that the developer completed for the application, these being figure 35 to figure 40. The figures show that all 3 pages were tested by the developer. The tests were given descriptions and reasons which better explains why those tests were required. The input and output were also tested to make sure that the tests were completed correctly, and the end results needed to complete the tests. The tests also had a pass and fail section that would quickly identify to the user if a section failed the tests. The tests all have numbers to easily identify the tests to the developer.

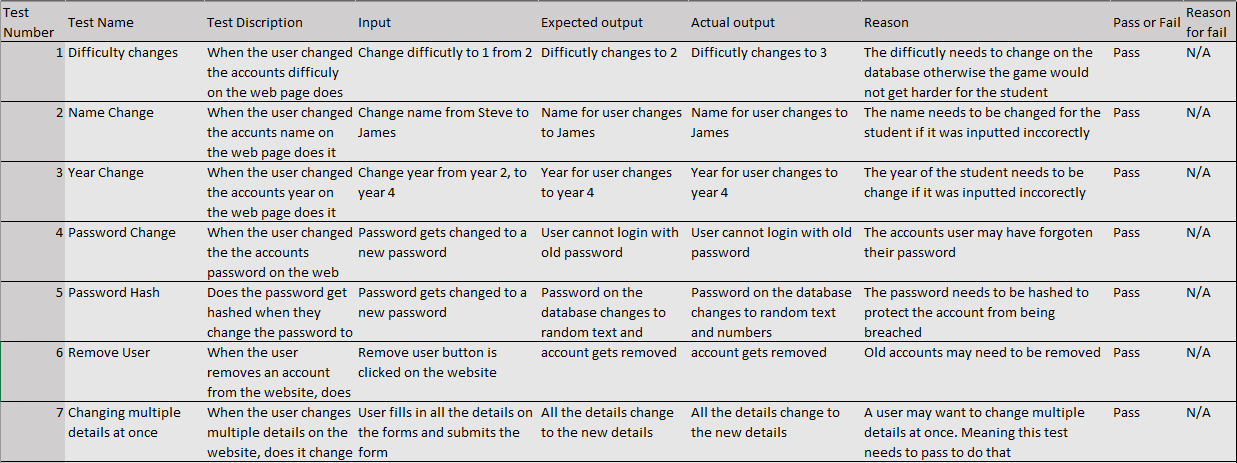


Figure 25: Database testing table

Figure 25 shows the testing that was completed on the database server side of the project. Some of the test related to the database were included in other sections. The figure shows the different tests that were completed to make sure the database updated information correctly with the website with no issues. The issues in this section would relate to the other pages within the website. However, without changes on the database, the website will stay the same and the above tests are carried out to ensure those changes take place.

# Critical Evaluation

We are of the opinion that the project was successful with everything that was planned completed as intended. However, we did not consider a potential world pandemic at the time of planning, neither did we expect a global lockdown that would restrict the ability to run the studies.

The answer buttons could be improved by utilising canvases instead of buttons, since buttons on their own are rather boring to use to a young person. While buttons do offer the same functionality, canvases are more interactive because they can allow for changes such as visual changes, for example, animations can be added to the pages [28], such as visual effects that follow the mouse.

We are also of the opinion that had we completed more research and considered more and varied prototypes using different navigation we could have developed a more attractive website for the targeted audience of primary age students and teachers. Completion of such work would have facilitated more focussed designs to be created.

Creating only two protypes limited our work and a wider range of investigations would have highlighted more issues for us to consider in the development of the website

We felt that utilising Atom, Git and GitHub was useful. GitHub desktop was very good at handling and Git and that Atom worked very well as an IDE. However, we feel that we should have run a locally hosted server for development as we found that using Aberystwyth was slow and difficult to use due to not having the ability to see errors in terminal. While PHP cannot be run on windows locally, a sever could have been set up on the local host that would have allowed for testing without connecting to the VPN.

To summarise If we were to design the website again, we would use canvases instead of buttons. This can be changed in future revision of this work. We would also like to change how players are scored and would like to do a dynamic scoring based on the difficulty of the answer (such as a 2 significant number question would give double points compared to a single significant figure question).

We think that the websites achieves its aims because it does complete all of the aims and objectives that where set at the start of the project, with only the testing needing to be changed to a different focus group. The focus group however had to be changed due to reasons out of control of the developer, Covid-19.

#### Future Work

At the end of the first draft, the project still has the potential to improve by adding new sections. However, these sections might have been too complex or too large to be completed by the deadline of the first draft but they still have the potential to be completed by a future date.

###### Different game types

One of these will be the spelling section, which as discussed earlier, would have involved a large and complex section of the project. If this section was added it would have been less likely that the project would have been completed by the deadline.

###### Adding students via the website

Another future project that could be included in the work would be the addition of allowing students to be added to the database through the website rather than through the external admin. This would involve creating a new page which would allow the creation of a new school and users within the school, from students to admin. This would also allow admins the option of adding new students to pre existing schools. However, this would need to be thoroughly tested regarding security and data protection.

###### Canvases

More future work would include changing the buttons that a user clicks form buttons and into canvases. This would not be a major piece of work, as the details for the canvases are already set out in an array. However, these canvases would not just be bland, they would need to have visual stimulation animations when they are clicked on by the user.

# Annotated Bibliography

[1] T. P. Group, “History of PHP,” Group, The PHP, 25 January 1998. [Online]. Available: https://www.php.net/manual/en/history.php.php. [Accessed 07 04 2020].

This Page shows the age of PHP and is the official page of PHP

[2] W. Team, “WEBASSEMBLY COMMUNITY GROUP,” W3C, 29 April 2015. [Online]. Available: https://www.w3.org/community/webassembly/. [Accessed 08 05 2020].

This page is the official page of WebAssembly and shows its age

[3] K. Padia, “Is Node.js really better than PHP? Let’s go Through the Pros and Cons,” datafloq, 2 August 2018. [Online]. Available: https://datafloq.com/read/is-nodejs-better-than-php-pros-cons/5297. [Accessed 07 May 2020].

This page compares Node.js to PHP, contains pros and cons for both

[4] “WebAssembly,” Mozilla, 8 February 2020. [Online]. Available: developer.mozilla.org/en-US/docs/WebAssembly. [Accessed 2020 May 07].

The official page for WebAssembly

[5] “Index,” Photon Storm Ltd, 2020 . [Online]. Available: https://phaser.io/. [Accessed 01 May 2020].

Official page for the phase framework

[6] S. Ltd, “Index,” SpryMedia Ltd, 2007. [Online]. Available: https://datatables.net/. [Accessed 10 May 2020].

Offical page for DataTables

[7] BBC, “'Zoombombing' targeted with new version of app,” BBC, 23 April 2020. [Online]. Available: https://www.bbc.co.uk/news/business-52392084. [Accessed 09 05 2020].

A publication about the new 'Zoombombing' attacks

[8] N. S. Team, “The Cross-site Scripting (XSS) Vulnerability: Definition and Prevention,” Netsparker, 18 April 2019. [Online]. Available: https://www.netsparker.com/blog/web-security/cross-site-scripting-xss/. [Accessed 07 05 2020].

A publication about Cross-site Scripting and ways to defend against it.

[9] K. R, “Living with password re-use,” The National Cyber Security Centre, 24 August 2017. [Online]. Available: https://www.ncsc.gov.uk/blog-post/living-password-re-use. [Accessed 05 May 2020].

Publication about the security issues with re-using the same passwords across multiple sites

[10] P. BROWN, “Why do you need to Salt and Hash passwords?,” Culttt, 21 Januray 2013. [Online]. Available: https://culttt.com/2013/01/21/why-do-you-need-to-salt-and-hash-passwords/. [Accessed 08 05 2020].

A publication about password hashing and salt

[11] T. P. Group, “password\_verify,” The PHP Group, No Date. [Online]. Available: https://www.php.net/manual/en/function.password-verify.php. [Accessed 08 04 2020].

Documentation page for the password\_verify function.

[12] GNU, “GNU General Public License,” GNU, 2016. [Online]. Available: https://www.gnu.org/licenses/gpl-3.0.en.html. [Accessed 20 05 2020].

Official page of the GNU General Public License

[13] phpMyAdmin, “Bringing MySQL to the web,” phpMyAdmin, 1998. [Online]. Available: https://www.phpmyadmin.net/. [Accessed 30 April 2020].

Official webpage for PhpMyAdmin

[14] G. TeamViewer AG, “index for TeamViewer,” TeamViewer AG, Germany, No Data. [Online]. Available: https://www.teamviewer.com/en/. [Accessed 24 March 2020].

Official website for TeamViewer

[15] W3schools, “CSS Flexbox,” Refsnes Data, 1999. [Online]. Available: https://www.w3schools.com/css/css3\_flexbox.asp. [Accessed 02 April 2020].

Documentation for CSS Flexbox

[16] J. B. a. others, “index of Kahoot,” Kahoot, March 2013. [Online]. Available: <https://kahoot.com/>.

Official Kahoot website

[17] “Index of Topmarks,” Topmarks Online Ltd, 1998. [Online]. Available: https://www.topmarks.co.uk/. [Accessed 04 May 2020].

Official Topmarks website

[18] Google, “Manage teaching and learning with Classroom,” Google, No Date. [Online]. Available: https://edu.google.com/products/classroom/?modal\_active=none. [Accessed 2020 May 07].

Official google classrooms website

[19] N. Raut, “What is the difference between session and cookies?,” tutorialspoint, 31 January 2018. [Online]. Available: https://www.tutorialspoint.com/What-is-the-difference-between-session-and-cookies. [Accessed 09 05 2020].

Publication about the difference between sessions and cookies

[20] C. DRUMOND, “Scrum: Learn how to scrum with the best of ‘em,” Atlassian, 'no date'. [Online]. Available: https://www.atlassian.com/agile/scrum. [Accessed 08 May 2020].

Publication about the scrum methodology

[21] M. REHKOPF, “Sprints,” Atlassian, 'No Date'. [Online]. Available: https://www.atlassian.com/agile/scrum/sprints. [Accessed 2020 May 08].

Publication about scrum sprints and information about how they work

[22] GitHub, “GitHub,” GitHub, No Date. [Online]. Available: https://github.com/. [Accessed 05 May 2020].

Official GitHub website

[23] GitHub, “Index of Atom text editor,” GitHub, 24 April 2019. [Online]. Available: https://atom.io/. [Accessed 01 05 2020].

Official website for the ATOM IDE

[24] Mozilla, “Math.random(),” Mozilla , 28 April 2020. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Math/random. [Accessed 10 05 2020].

Documentation for the Math.random() function

[25] S. Tatham, “Download PuTTY: latest release (0.73),” Simon Tatham, 29 09 2019. [Online]. Available: https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html. [Accessed 2020 May 01].

Official website for the putty application

[26] Selenium, “Selenium,” Selenium, 2010. [Online]. Available: https://www.selenium.dev/. [Accessed 10 05 2020].

Official website for Selenium, a third party unit testing software

[27] OpenJS, “Index of Mocha,” OpenJS, No Date. [Online]. Available: https://mochajs.org/. [Accessed 03 May 2020].

Official website for Mocha, a third party unit testing software

[28] Mozilla, “Basic animations,” Mozilla , No Date. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Canvas\_API/Tutorial/Basic\_animations. [Accessed 07 05 2020].

Website about creating basic animations within JavaScript canvases

# Appendices

## Tests

#### User testing document

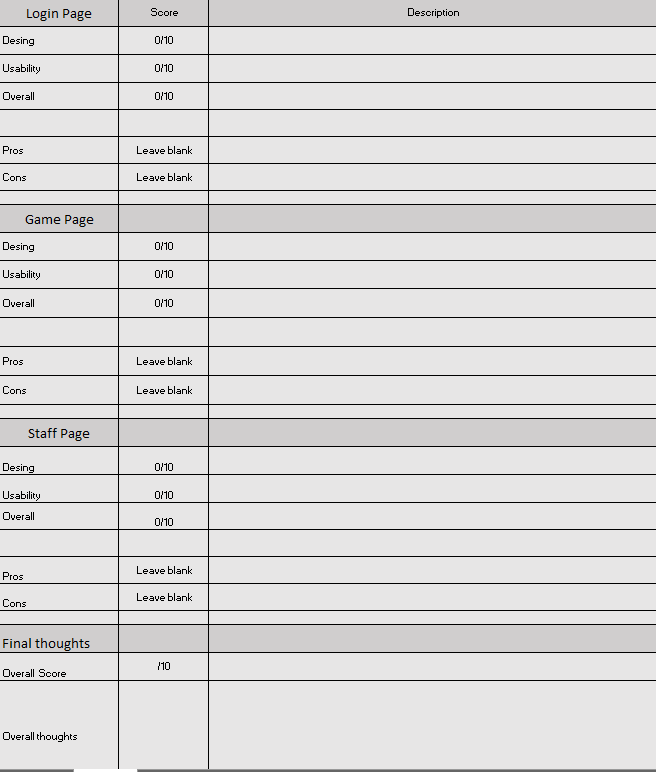


Figure 26: Blank document that was given to other users for user testing



Figure 27: User test one



Figure 28: User test two

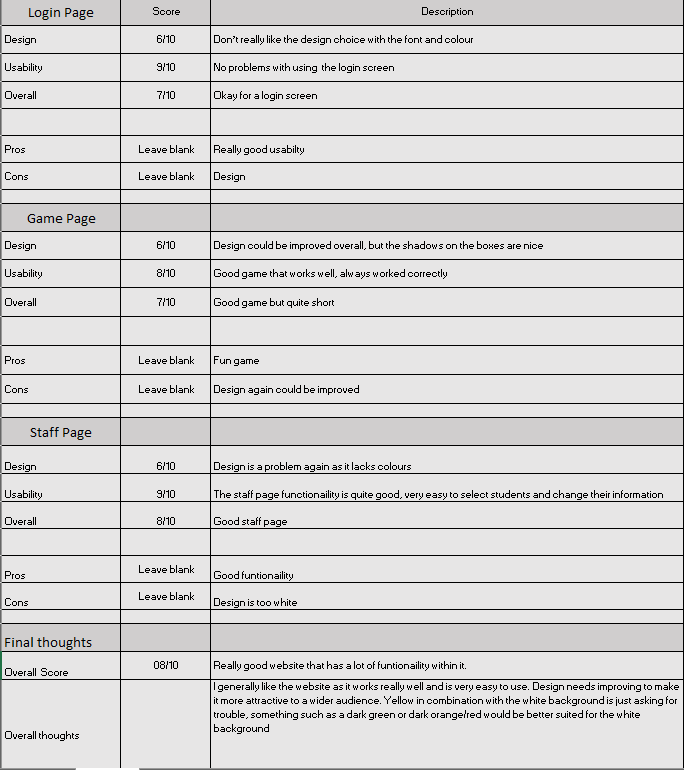


Figure 29: User test three

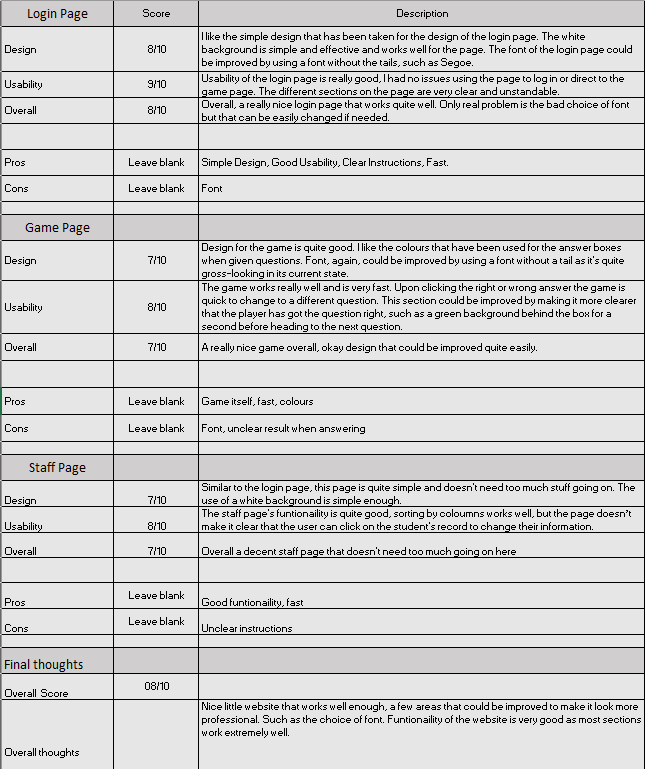


Figure 30: User Test four

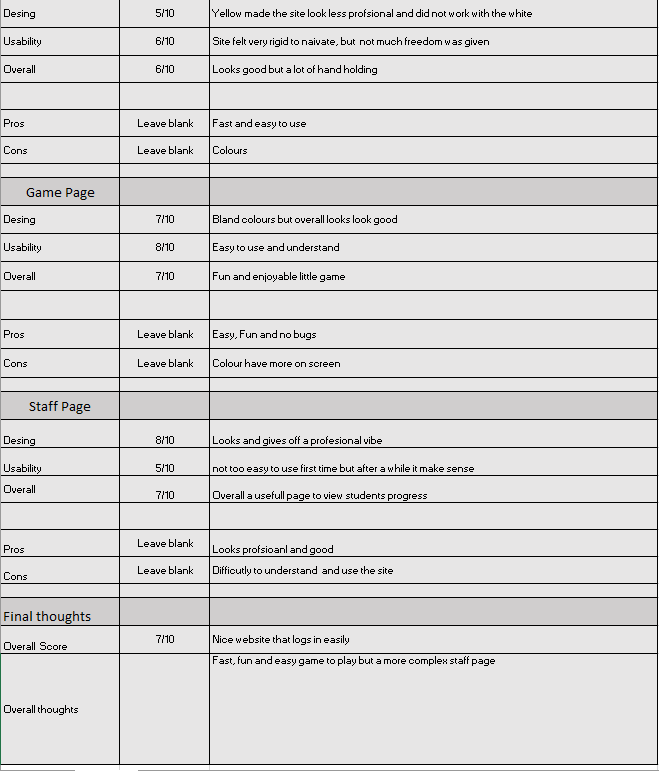


Figure 31: User test five

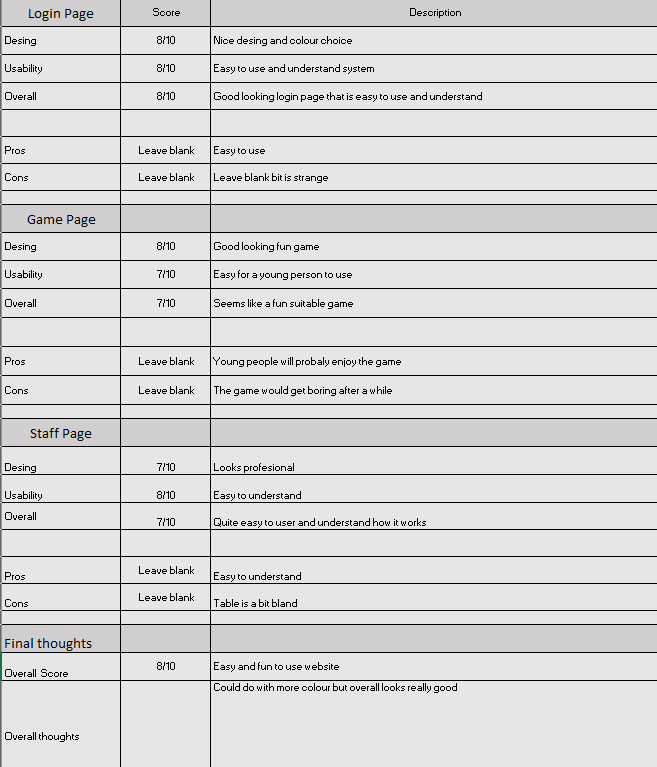


Figure 32: User test six



Figure 33: User test seven

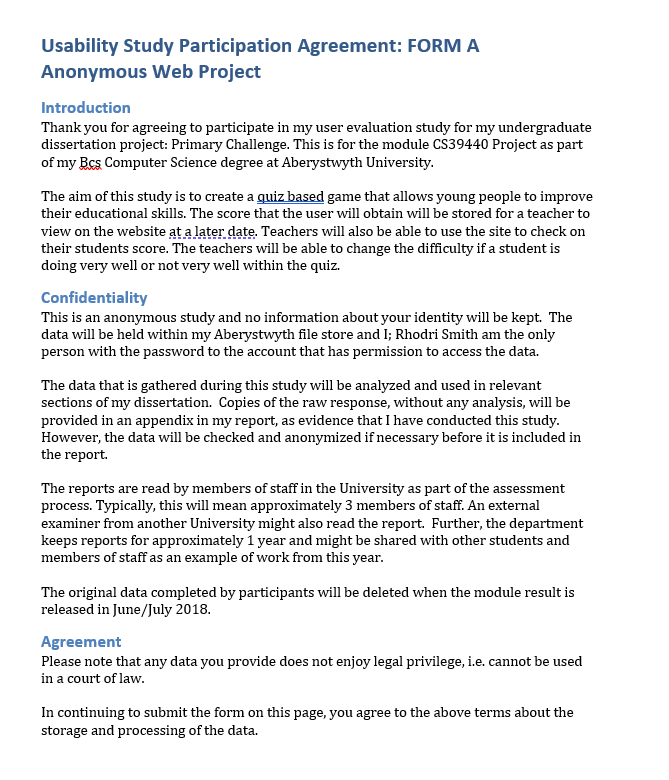


Figure 34: Form given to user to state their data will be kept safe and anonymous

#### Login Page tests

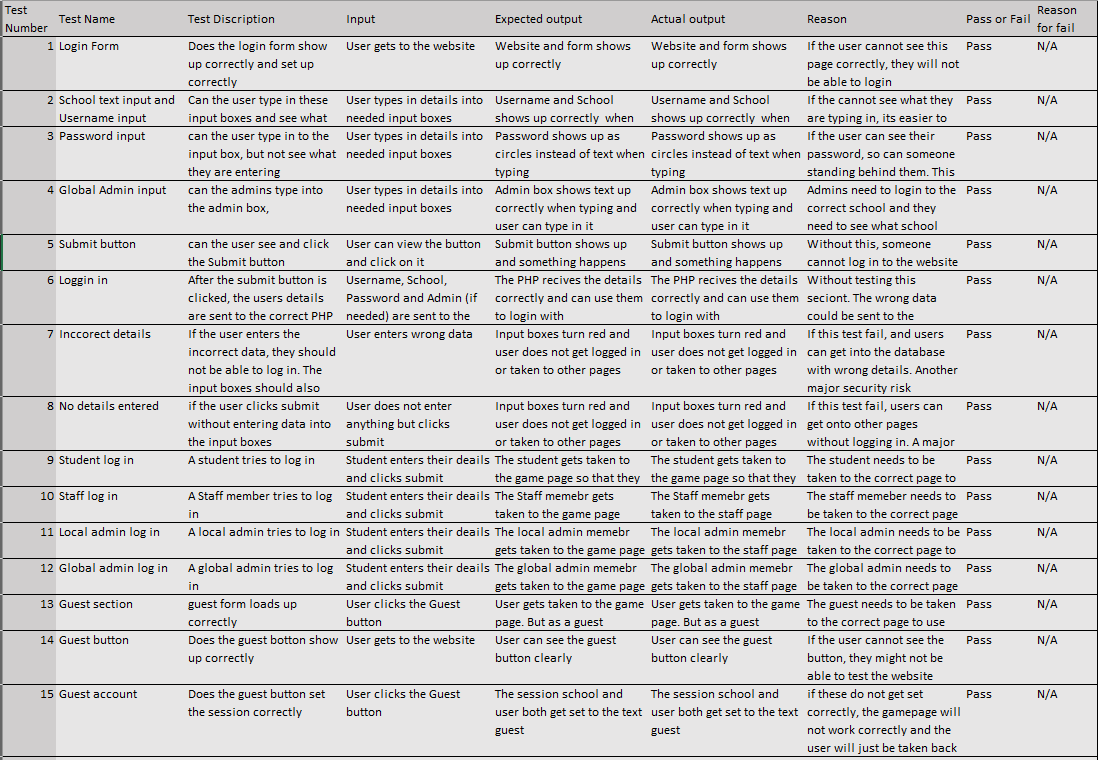


Figure 35: Login Page Manual Testing Page 1



Figure 36: Login Page Manual Testing Page 2

#### Game Page test

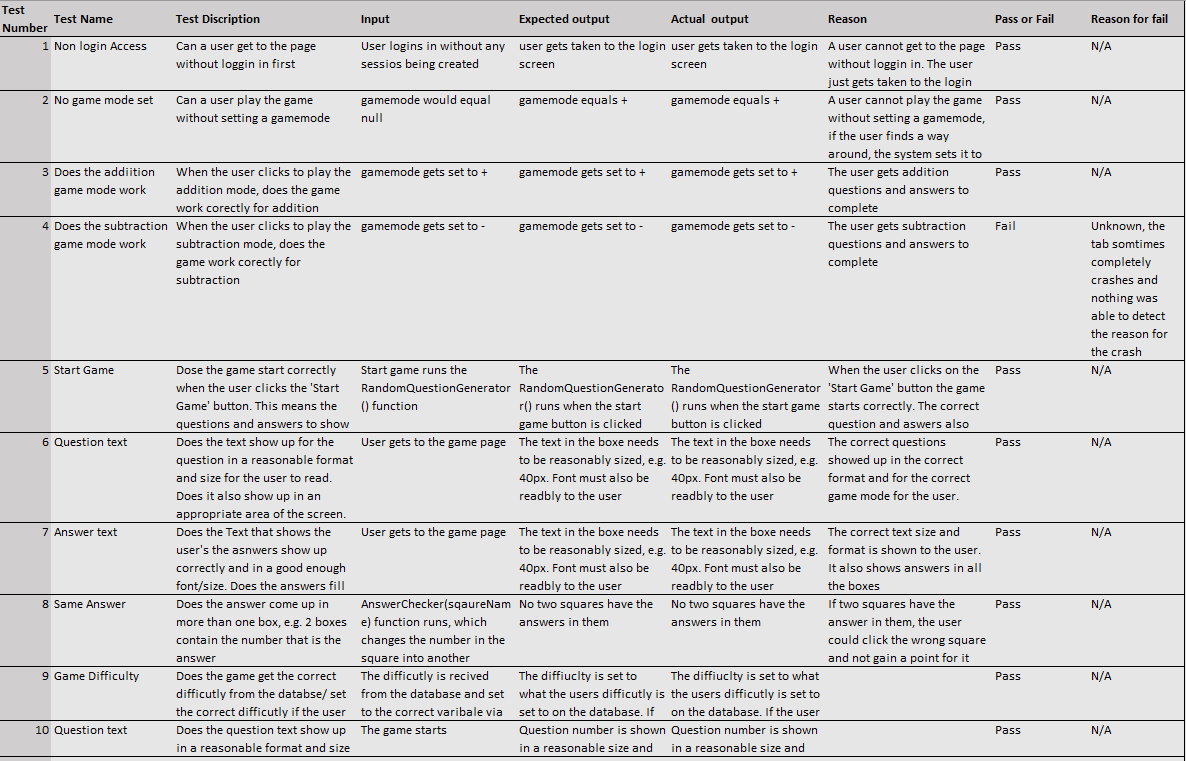


Figure 37: Game Page Manual Testing Page 1

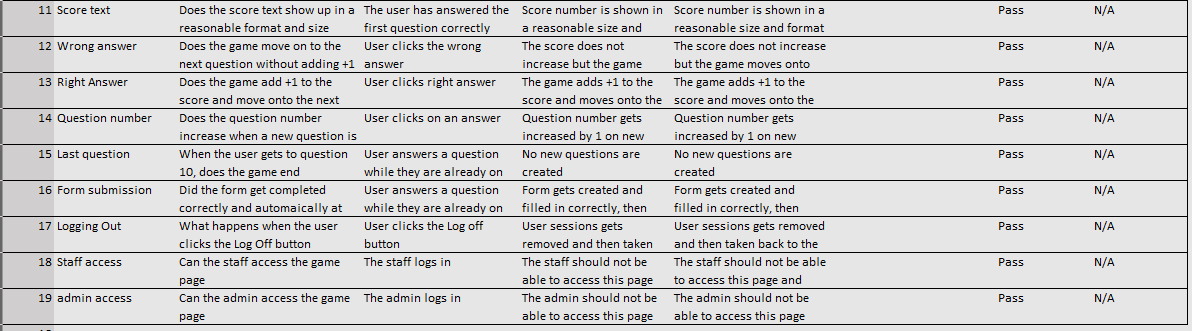


Figure 38: Game Page Manual Testing Page 2

#### 

#### Staff Page test

#### 

Figure 39: Staff Page Manual Testing Page 1

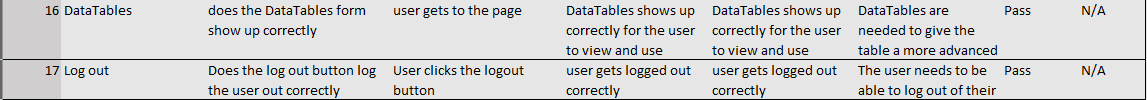


Figure 40: Staff Page Manual Testing Page 2

## Diary

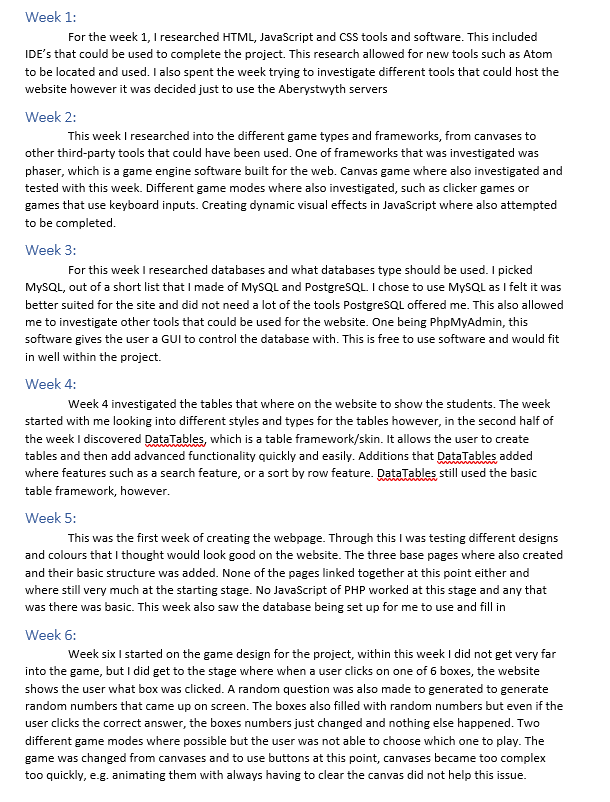


Figure 41: First page of the diary document

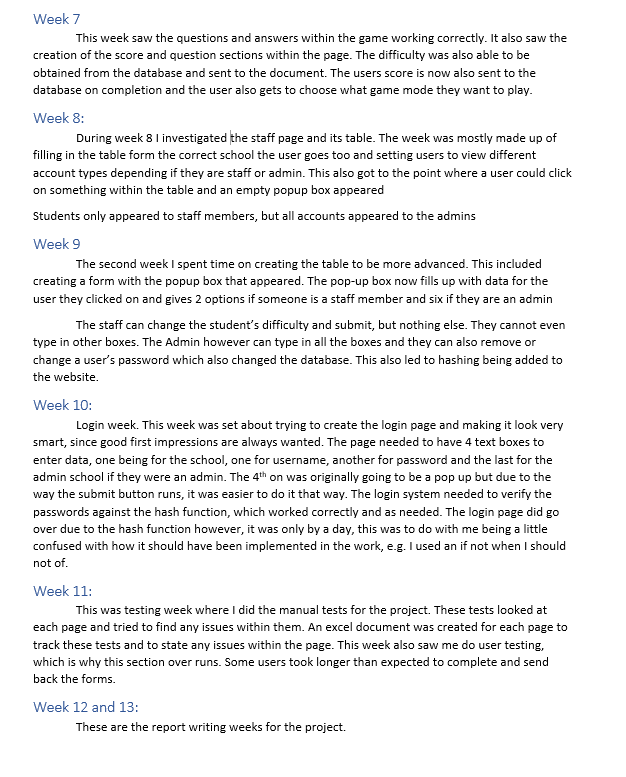


Figure 42: Second page of the Diary document

## Design and Prototype

#### Login Page

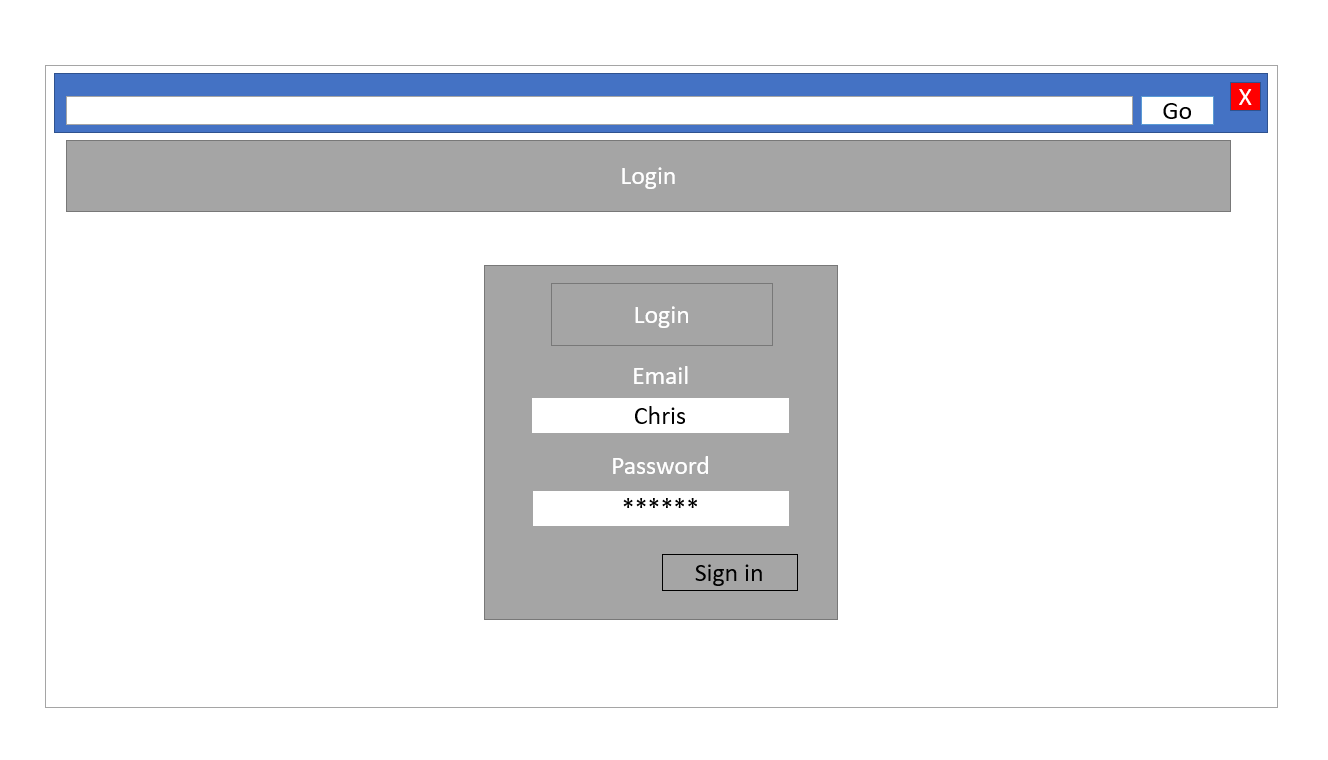


Figure 43: Prototype of Login page

#### Staff Page

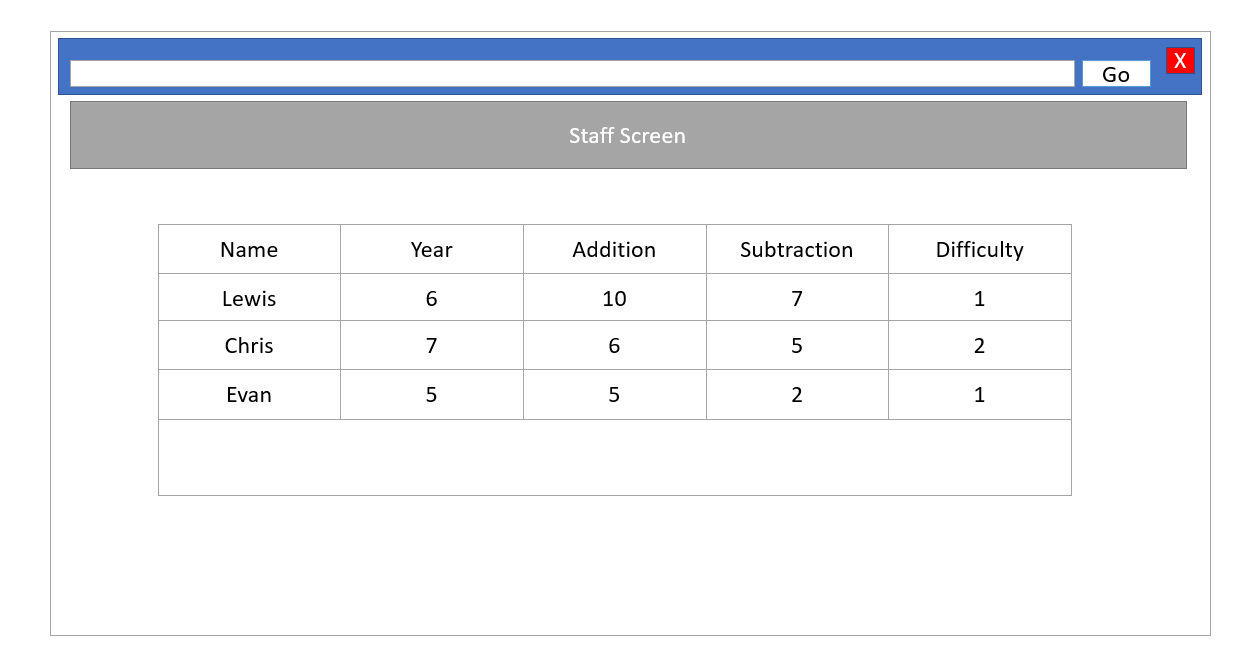


Figure 44: Prototype of Staff Page

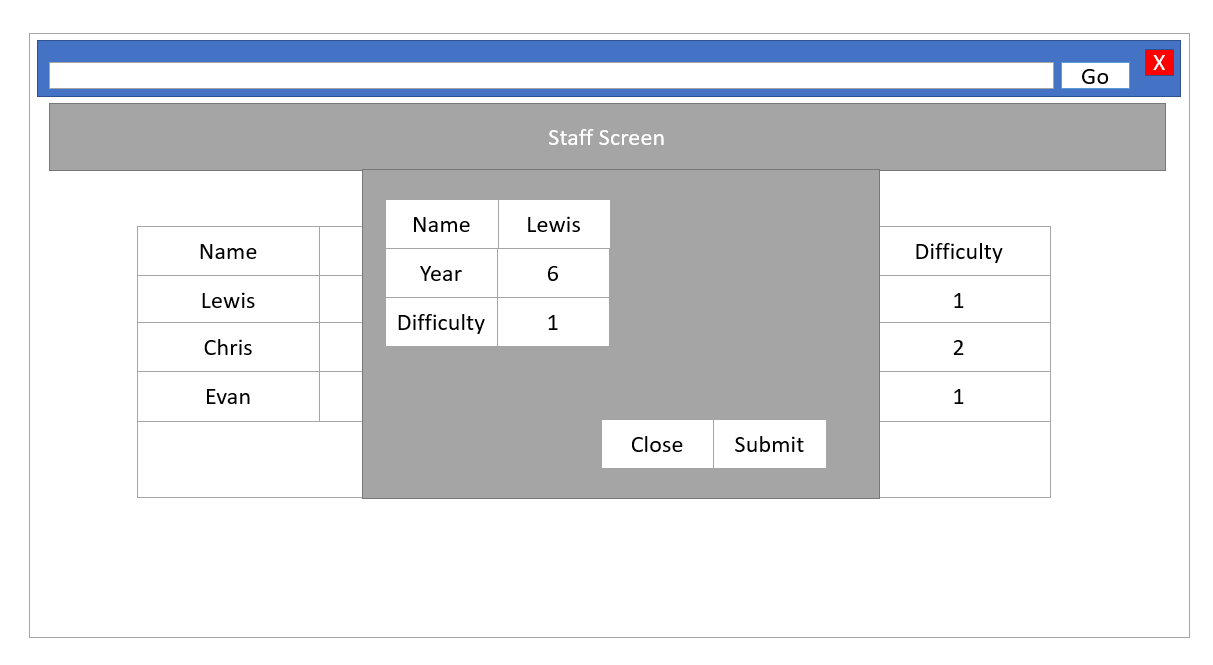


Figure 45: Prototype of staff page and pop-up box

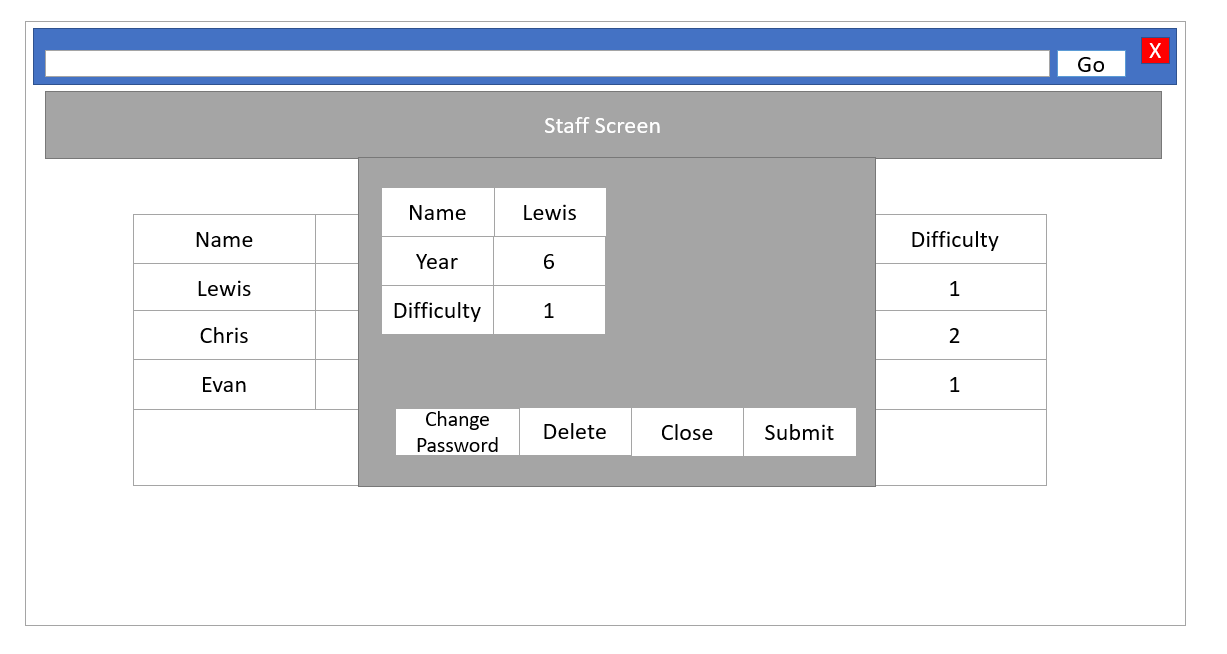


Figure 46: Prototype of what the admin will see on the staff page pop-up box

#### Game Page

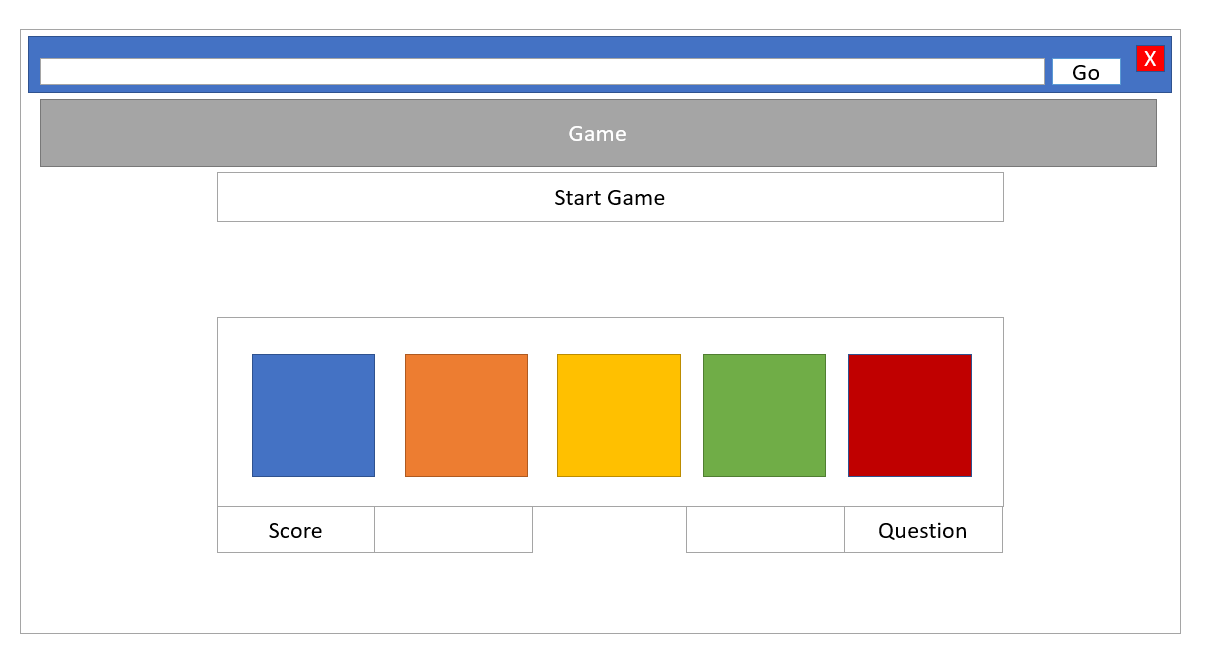


Figure 47: Prototype of the Game Page

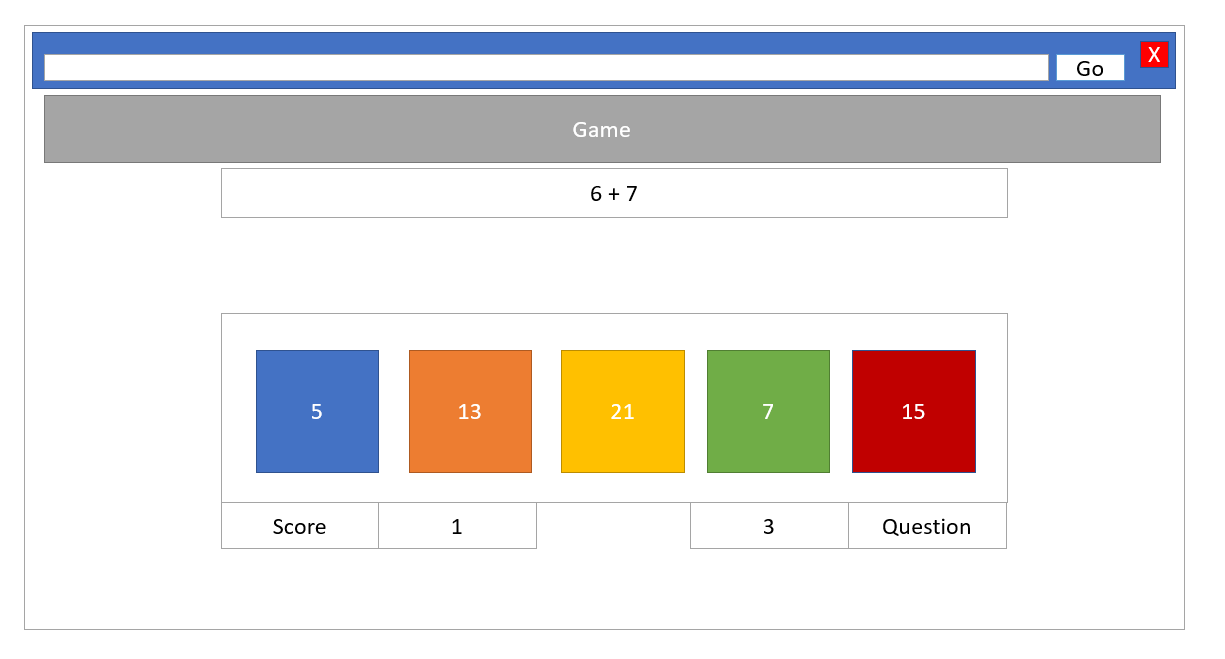


Figure 48: Prototype of Game Page during game

## Project Goals Document

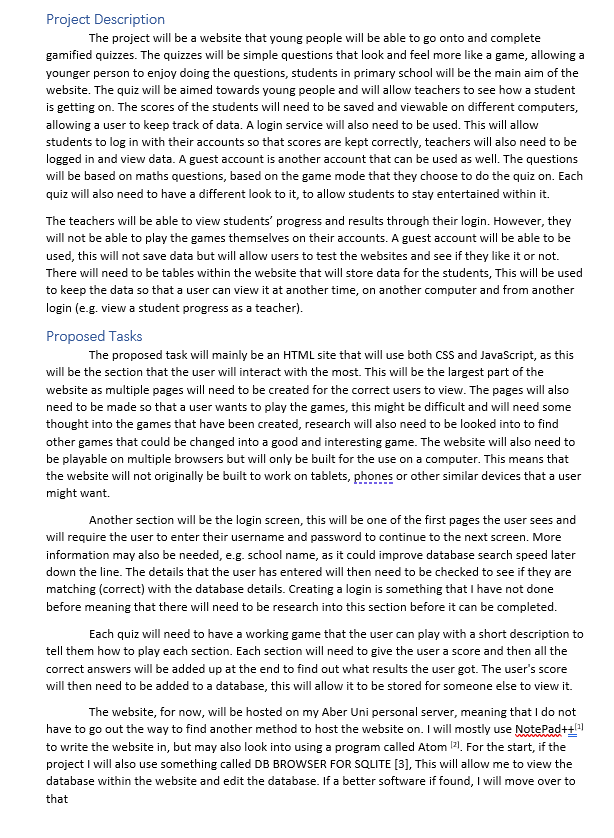


Figure 49: Page One of The Project Goals Document

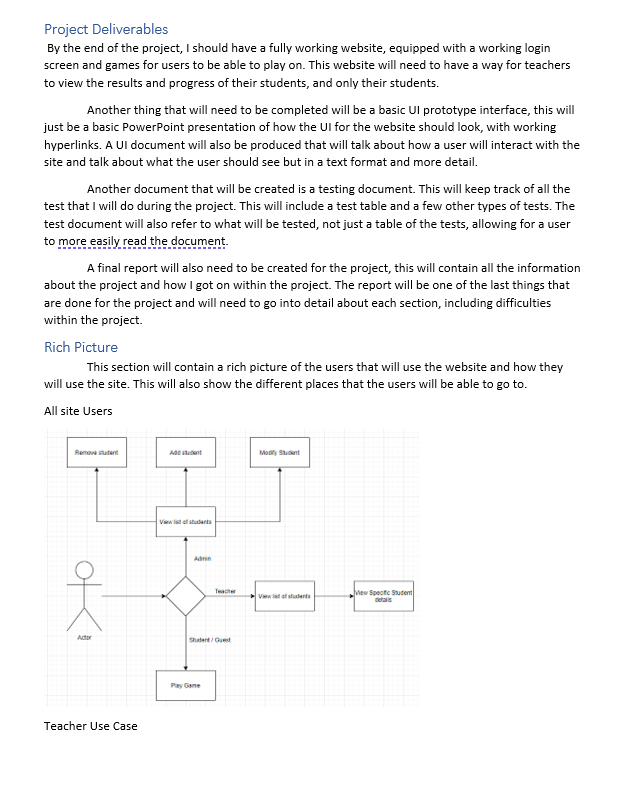


Figure 50: Page two of the project goals document

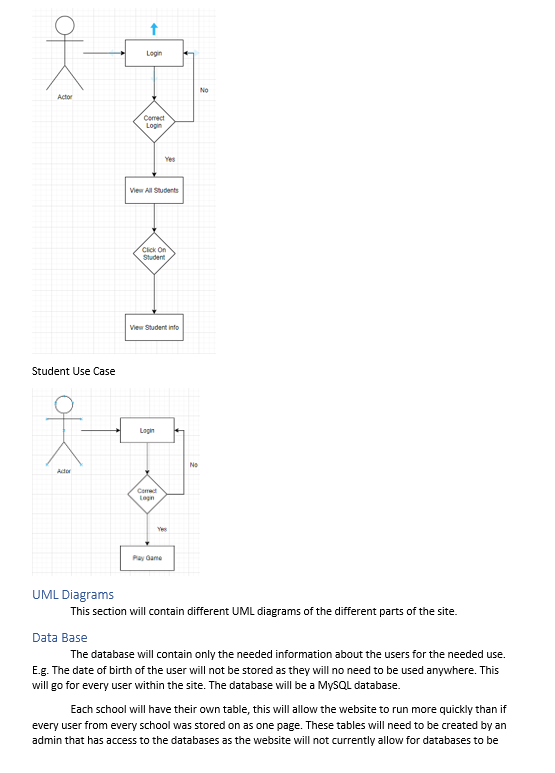


Figure 51: Page three of the Project Goals Document

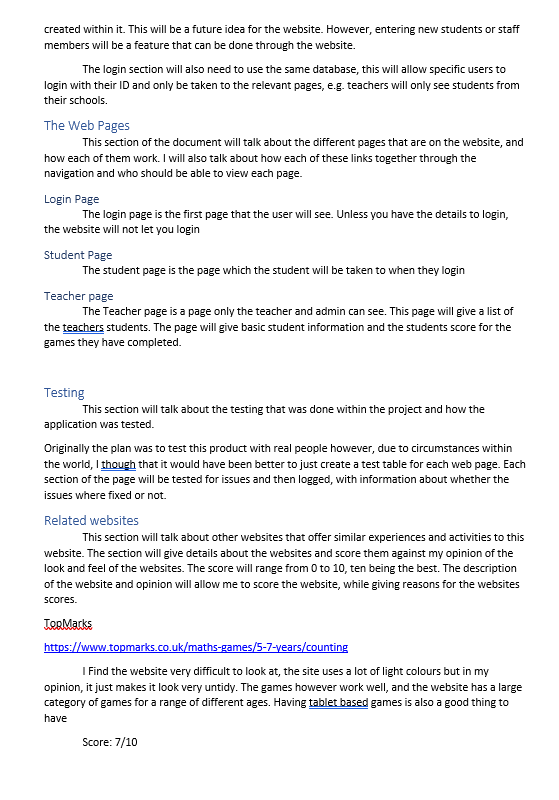


Figure 52: Page four of the Project Goals Document

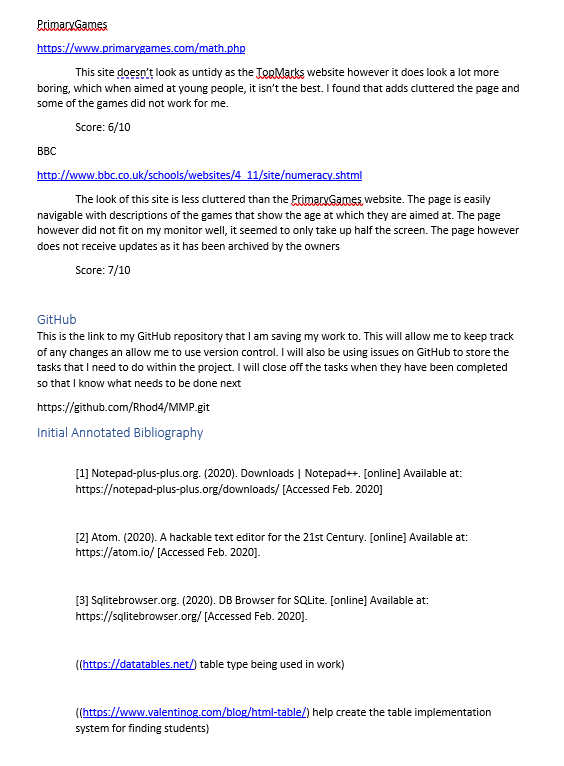


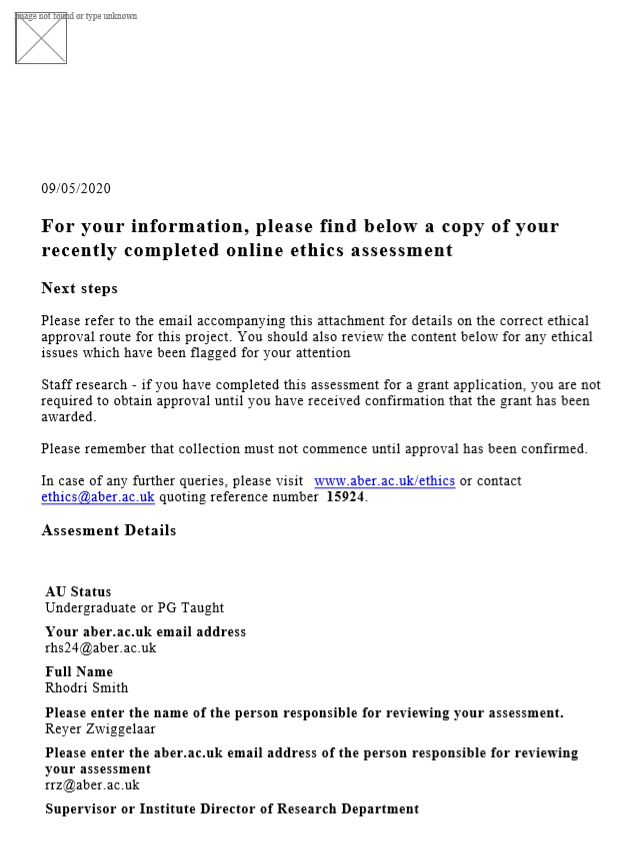
Figure 53: Page Five of the Project Goals Document

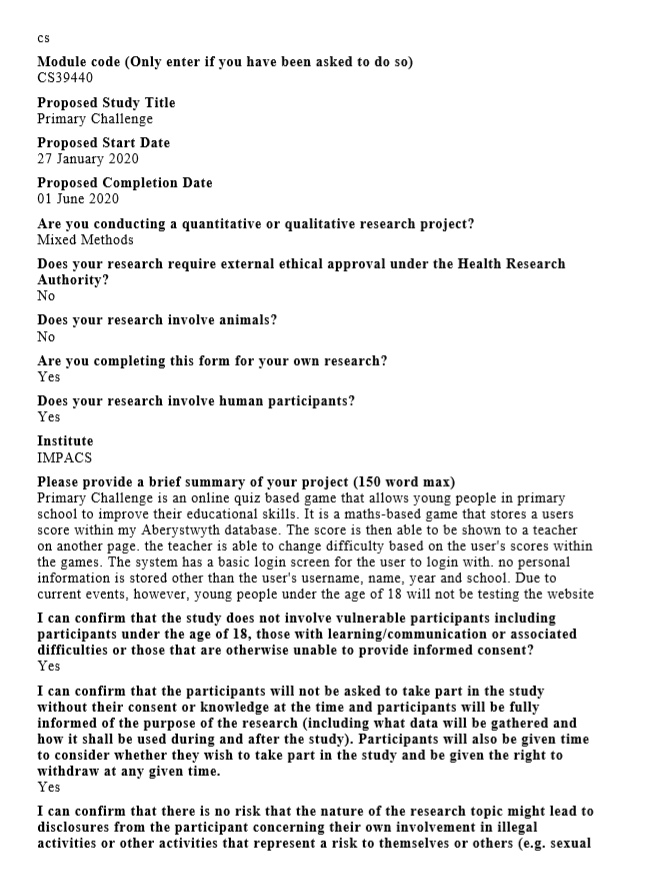
* 1. Third-Party Code and Libraries

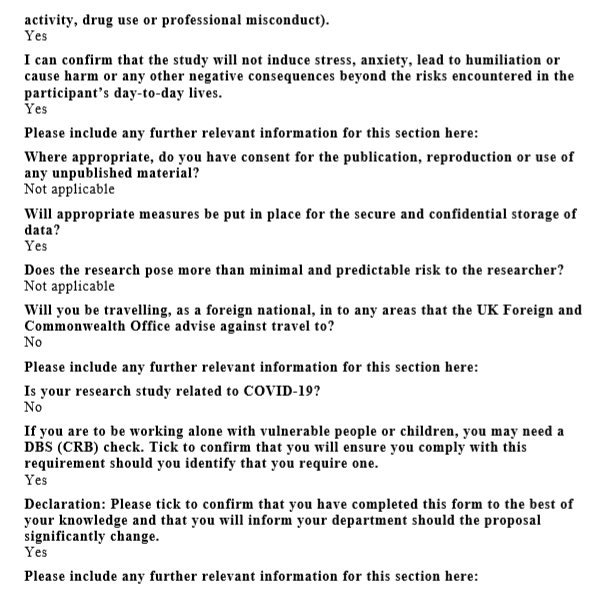
DataTables [6] is third party software that was used within the application to show a table in a more advanced and useful way. DataTables is a framework that allows a user to create professional working tables with ease. DataTables has a MIT license, which allows a user to use the software free of charge to a user. This licence allows a user to use the application freely with zero restrictions. For more information on the DataTables licence, a user can go to the DataTables licence page (<https://datatables.net/license/mit>).

PhpMyAdmin [13] is another third party software that was used. It is a free to use under the GNU General Public License [12]. This means that it is free for a user to have to control and maintain their database with. PhpMyAdmin gives an advanced GUI for the user to be able to use, stopping the need of the user to use the command line to control the database. A link to the license agreement can be found here (<https://www.phpmyadmin.net/license/>)

* 1. Ethics Submission







* 1. Code Samples

1. https://atom.io [↑](#footnote-ref-2)