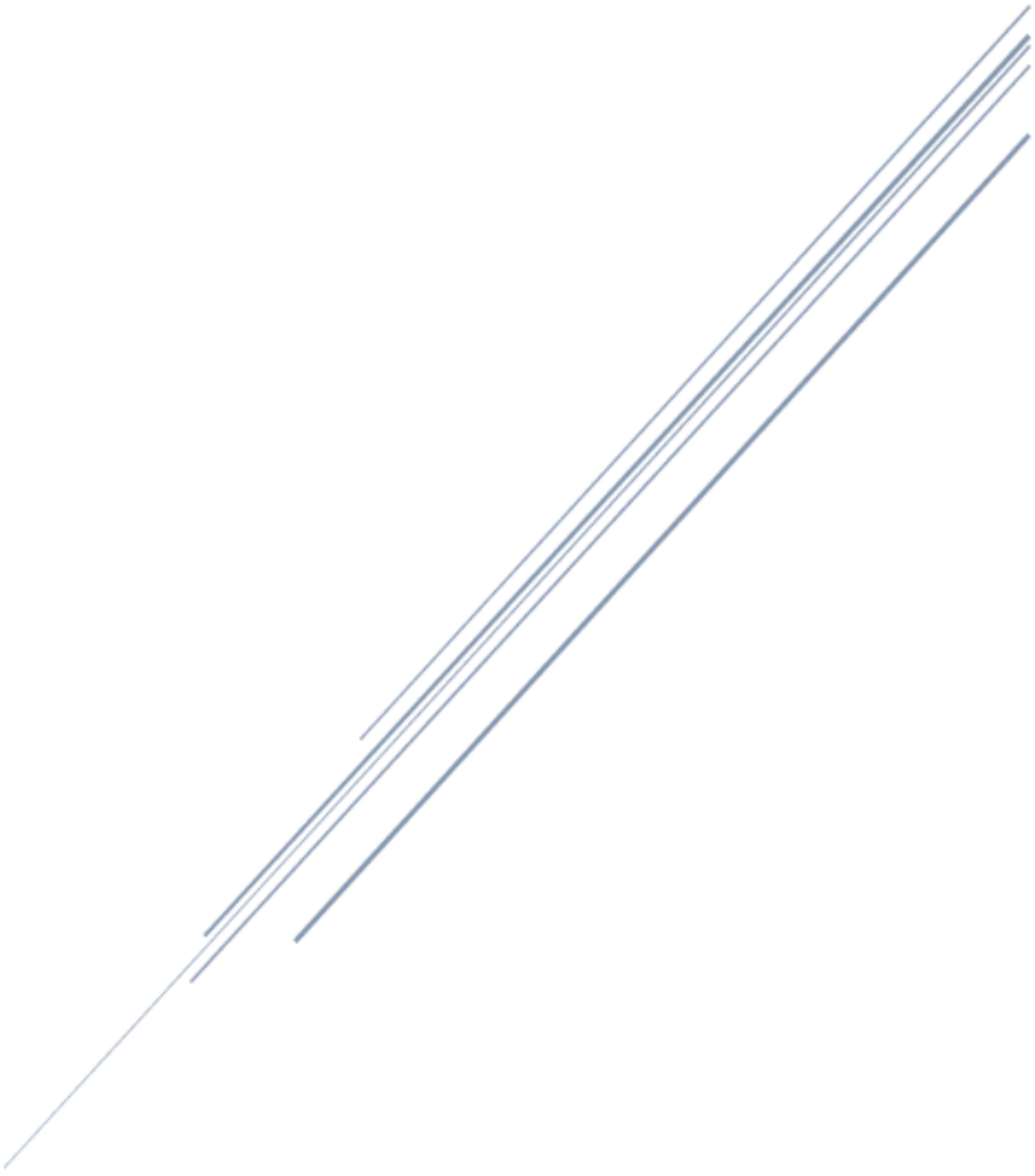
Social Engineering Training Tool



Michael Green, P2418088

CTEC3451 – Development Project (Final Deliverable)

Module Leader: Dr Hossein Malekmohamadi

Supervisor: Nick Ayres

Table of Contents

1. Introduction 4
1.1 The Problems and Objectives 4
2. Major Components 5
2.1 Quizzes 5
2.2 Page Switching 9
2.3 Graphical User Interface 12
2.3.1 Background Information 12
2.3.2 Buttons 12
2.3.3 CSS, HTML and JavaScript Integration 15
2.3.4 Score and Current Question Counters 15
3. Development Lifecycle 16
3.1 General Background 16
3.2 Implementation 17
3.2.1 Planning 18
3.2.2 Building 19
3.2.3 Testing 20
3.2.4 Reviewing 20
4. Reflection and Critical Analysis 20
4.1 Strengths 20
4.2 Challenges 21
4.3 In Hindsight 21
4.4 Software Tools 22
4.4.1 Visual Studio Code 22
5. Conclusion 23
6. References 24
7. Appendices 26
Appendix A 26
Planning 26
Testing 36
Appendix B 41
Application: 41

Table of Figures

Figure 1 - Basic Quiz design displaying the three elements 6
Figure 2 - Multiple Choice assessment 7
Figure 3 - True or False Quiz code that saves score 7
Figure 4 - JavaScript code for navigation to results screen 8
Figure 5 - JavaScript code that randomises the question order 8
Figure 6 - Results Page 9
Figure 7 - Page Switching functionality 9
Figure 8 - Navigation Bar HTML Code 10
Figure 9 - Page Navigation Code 10
Figure 10 - Logical Flow Overview 11
Figure 11 - Progress Bar 11
Figure 12 – Navigation Bar HTML Code 12
Figure 13 - Navigation Bar Styling Code 13
Figure 14 - Example of Hyperlinked Text 13
Figure 15 - Hyperlink HTML Code 13
Figure 16 - Code for User Answer Selection 14
Figure 17 - CSS Linked into HTML Page Figure 18 - JavaScript Linked into HTML Page 15
Figure 19 - Score Counter JavaScript Code 15
Figure 20 - Sprint Stages 17
Figure 21 - The Scrum Framework (Darwish, N.R and Megahed, 2016) 17
Figure 22 - Kanban Board Example 19
Figure 23 - Building Process 19
Figure 24 - Visual Studio Code 23

1. Introduction

Since the new millennia, social engineering attacks have not only increased, but also proved difficult to prevent and mitigate. Attacks are not only targeting large businesses, but also individuals, such as attacks in the financial sector on unaware or uneducated users even in surprising areas such as cloud services and social networks (Krombholz, Hobel, Huber and Weippl, 2015). The significance of this issue is shown as the cost of these attacks in the US is estimated to be $122 Billion, and these attacks often cause a greater societal impact then first recognised (Salahdine, Kaabouch, 2019). The attacks that are carried out are not only increasing, but also have a high success rate of 84% (Salahdine, Kaabouch, 2019). The following report will discuss and explore the design and implementation of a Digital Training Tool (DTT), produced to support users, and help mitigate attacks across the various social engineering vectors as effective tools are shown to increase user motivation in keeping themselves safe (Dincelli and Chengalur-Smith, 2020). Key aspects such as the GUI will also be discussed, as the key aspects should also be user friendly, as this will encourage engagement and overall functionality (Pirocca, Allodi and Zannone, 2020).

The below report will identify the problems and objectives of the DTT, the major components that created the end product, the Graphical User Interface (GUI) that provides the front end functionality and how this was designed. It will then identify the development lifecycle used, how the project was planned, built, and tested, along with reflection on what went well and what could have been done better in hindsight.

1.1 The Problems and Objectives

Since the dawn of the new digital age, DTTs have catered to a range of individuals, and allowed them to learn in a way that is engaging and sustainable. Consequently, these tools can be tailored to suit their preferred audience. From teaching foreign languages to training employees of an organisation, in order to protect themselves and the business by which they are employed. Furthermore, a DTT has typical features that enhance user learning, and also take advantage of their technological nature, in their inherent flexibility when compared to other forms, such as traditional classroom based learning. The combination of flexibility and user friendliness often creates a suitable and effective method of learning, which can be adapted to suit the business, user, or subject at hand.

The created tool that has been designed at the end of the development lifecycle, in the most basic form, is designed to provide easy to digest knowledge and engaging learning that can be translated into improving a typical users future actions on the internet or in wider life. Furthermore, the tool can take advantage of the technological benefits that the platform it is located on provides. This is that the tool can be accessed and used repeatedly, and also the user friendly nature that was a deliberate consideration from the tool's inception. However, a tool of this nature also has aspects of the design that have to be taken into consideration, particularly the aspects of the tool that provide interaction with the user, such as the Graphical User Interface or quizzes found within the tool.

Consequently, the tool has been designed to take advantage of the technological benefits it provides that would be either difficult to replicate, or simply unavailable with a traditional, in person framework such as classroom based, face to face learning. Although the tool is designed to be catered towards the lay person, the tool could also be used in a professional environment, as the additional training an employee may have had before the use of this tool

would only serve to enhance their progression and overall benefit of the tool. Additional learning before using a learning tool has also been proven to show enhanced benefits, (Anastasiadis, Lampropoulos and Siakas, 2021).

Throughout the development lifecycle, research was undertaken to understand and implement any specific information that could support in the design and implementation process of the training tool. The research undertaken was started during the project contract document, which can be found in the objective section. This section contained the research element that was used to influence design considerations such as general user enjoyment and usability, which would then be integrated to develop a DTT that was not only fit for purpose, but also successful in its aims. The research mentioned was also implemented into the first deliverable, found within the literature review section of the deliverable document. The research gathered and its benefits will be considered within particular aspects of the DTT, such as the user interface as this was a key aspect of the created tool in order to sustain engagement.

Therefore, the overall aim of this development project was to design, implement and deliver a DTT tool which educates users about a range of social engineer's attacks, in order to broaden their existing knowledge. A positive consequence of the creation of this tool would be that the user's future actions are taken with a higher degree of caution, which would subsequently reduce the chance of falling victim to the increasing number of cyber-attacks. During the creation of the tool, multiple aspects were pieced together to produce an effective training tool. These sectors can also be separated and analysed to identify how the tool was designed and implemented. These components are responsible for provide the key functionality that can be found within the tool. A key aspect is the Graphical User Interface (GUI), which provides the basic framework to allow the user to engage with other design aspects of the tool such as the quizzes, learning content and navigation. As a result, these components will be analysed, and discussed in detail to provide the logical reasoning behind each component, which will guarantee the tool has been created to an acceptable standard and allowed the project to achieve the previously set aims and objectives.

2. Major Components

2.1 Quizzes

As a key aspect of a training tool is feedback, the previously collated research was used in conjunction with additional design considerations to give prompt feedback, the DTT was designed to fulfil the feedback commonly found with previously existing training tools. This is prevalent particularly in the form of scoring. The quizzes are freely accessed, and are designed to provide instant feedback after the selection of an answer, in both the 'True or False' and 'Multiple Choice' quizzes. This method was implemented as an alternative to giving an overall score at the end, as the frequent feedback acts as a motivating factor, and also gives the user a level of transparency in regard to the effect their learning has had on their assessment and will then influence their future actions after using the tool. Furthermore, these assessments also allow for an alternate engagement with the tool, as there is a greater focus on input and feedback that would be considered more tactile in comparison to the learning areas within the tool. These quizzes also ensure user engagement remains intact and reduces the chance of the training tool having little educational value, and thus failing to achieve the original aims of the tool. The type of information and learning content displayed in the tool was designed to be aimed at the lay person, and consequently subtle

design alterations were implemented such as sub headings under mitigations in relation to social engineering attacks.

To begin an assessment, the user is first displayed with the prompt page, which confirms the user is ready to begin the quiz. The coding behind this functionality is the ‘ ‘ function which will be explained in more detail later in the report and is located within the page switching section**.** Furthermore, the buttons themselves are discussed in the GUI section**,** in particular the ‘a href’ code that is located within buttons that allow for effective navigation within the training tool.

Before the user is allowed to begin the quiz, the navigation path has been designed so that that the user is taken to a prompt page, before starting each quiz. These pages allow the user to confirm they are ready to start the quiz before they begin. They also allow the user to navigate back to the homepage using the ‘Home’ button that is cantered within the page. The quiz pages is typically comprise of three key features, which allow the assessments to function in their current format. The first is the question itself, that has been designed to be worded in a simple and intuitive way, with at times, a blank line inputted into the question to show there is a missing word to the statement required. Alternatively, there are statements that are either true or false, or in the multiple choice quiz’ case, at times they require the user to differentiate between similarly worded questions to ensure the user has confidence in their understanding of the answer in relation to the question. The second feature is the responsive elements that allow the user to receive feedback on their submitted questions, which are the available answers. The final element is the coding behind the end of quiz path, which identifies when the answers have been completed, and thus the quiz is over, and the score is saved. These components provide the core functionality required to allow the user to feed the knowledge that they have absorbed back into the system, which allows for a direct and transparent learning experience.

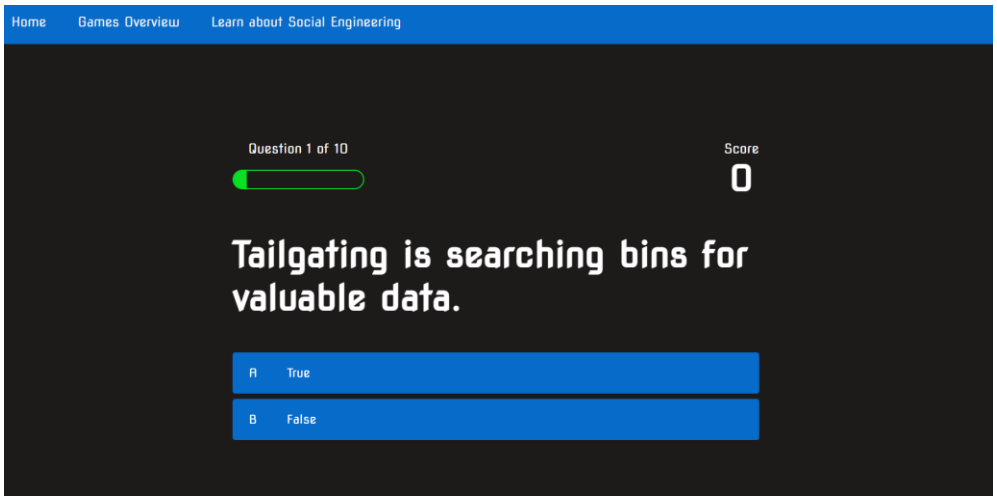


Figure 1 - Basic Quiz design displaying the three elements

The DTT was created using a combination of HTML pages which are the where the key interactive elements are created, Cascading Style Sheets which enable the styling for the created elements (CSS), and JavaScript which provides the logical functionality for the assessments. This is then designed and edited within Visual Studio Code which acts as a text editor in this scenario. The following quizzes located within the tool were designed and

assembled following the same format, with the rectangular clickable buttons that allow the user to select which answer they perceive to be correct. Figure 2 shows the multiple choice assessment, which allows the user to engage with the assessment at a swift pace, whilst also getting visible colour based feedback that is not only simple to understand when selected but is also a globally used method of signalling correct and incorrect. This can only serve to benefit the tool in its wider use in the future.

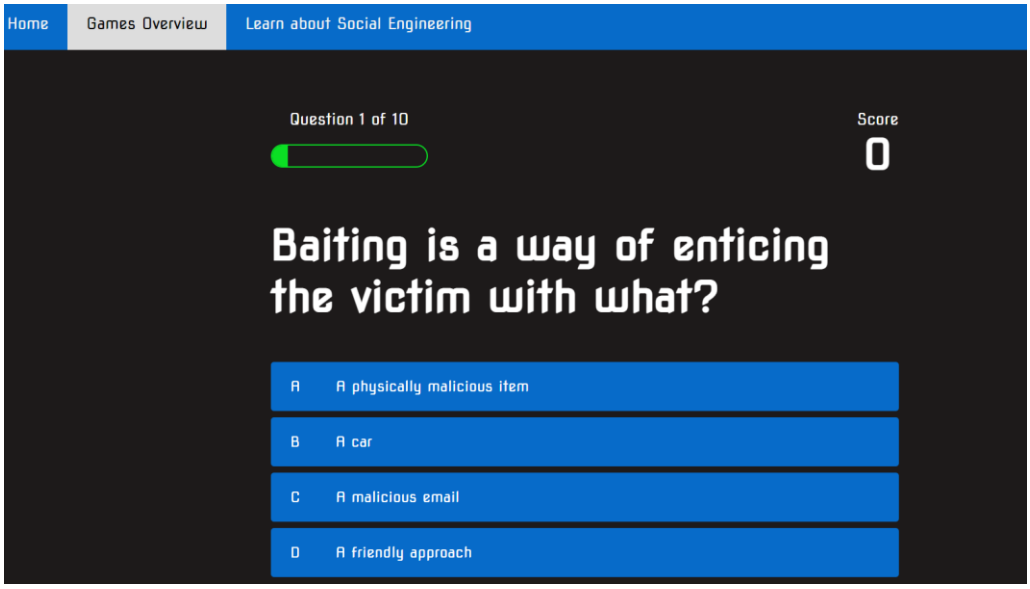


Figure 2 - Multiple Choice assessment

Continually, the complexity is differentiated between both types of assessment. The true or false quiz is designed to be simpler not only in form but also complexity of questions asked. This allows the user to begin to feel comfortable within the tool, and also gain confidence before they tackle the multiple choice quiz. Due to the higher difficulty, the user will need to continue to engage with the tool to allow for an increased success scoring in relation to the multiple choice quiz. This will ensure for a higher degree of engagement as the user progresses through their DTT navigation.

The designed process surrounding the user submitting their results and receiving scoring is done in two aspects. The first aspect is the real-time scoring the user receives as they being to answer the related questions. This can be seen in Figure 3, as the current score is promptly displayed above the current question that is being asked.

```
if(classToApply === 'correct') {
    incrementScore(SCORE_POINTS)
}
```

Figure 3 - True or False Quiz code that saves score

The method of completing the quiz and being navigated to submit and save the quiz score is called by the method of ‘A href’ within the JavaScript code. This essentially puts the questions in a loop, until it is recognised that the question is beyond the total question count. Figure 4 shows the code that navigates the user towards the results screen.

```
getNewQuestion = () => {
    if(availableQuestions.length === 0 || questionCounter > MAX_QUESTIONS) {
        localStorage.setItem('mostRecentScore', score)
        return window.location.assign('/C:/Users/micha/OneDrive/Documents/Uni%20Work/Uni%20Work/Final%20Year/FYP/HTML/Multiple%20Choice%20Results.html')
    }
```

Figure 4 - JavaScript code for navigation to results screen

The key aim of the return window. Location code is to firstly check that the question limit of ten has been completed, and if so to then navigate the user automatically to the results screen. The lack of a 'submit' button also prevents the user from attempting to submit their score before all questions have been attempted and is a simple yet effective way of controlling the scoring process. This would also prevent cheating, as the user cannot submit their results, and then go back and view the question again in an attempt to get the answer correct on their revised attempt.

Although there is a fixed set of ten different questions in each assessment, the questions are randomised each time the quiz is loaded. The benefit of this is that it prevents cheating, as it would be more difficult for the user to memorise the questions, as can often be done in a pattern with traditional style quiz based assessments. Figure 5 shows the JavaScript code that randomises the questions.

```
const questionsIndex = Math.floor(Math.random() * availableQuestions.length)
currentQuestion = availableQuestions[questionsIndex]
question.innerText = currentQuestion.question
```

Figure 5 - JavaScript code that randomises the question order

Once the user has completed all ten questions, the user is then prompted to save their score. This feeds into the leader board functionality that has been implemented into the tool. As the DTT was designed to be for the lay person, this encourages friendly competition, and also an increased level of engagement over time with the training tool. As users are encouraged to attempt the quizzes more than once, this will also increase their memorisation of the information they have absorbed whilst using the tool, and further enhancing the effectiveness of the tool.

After the user has saved their score, they can either retry the quiz in an attempt to improve their score, or they can be redirected to the homepage. Figure 6 shows the results page, and the options that the user has to navigate to beyond this point.

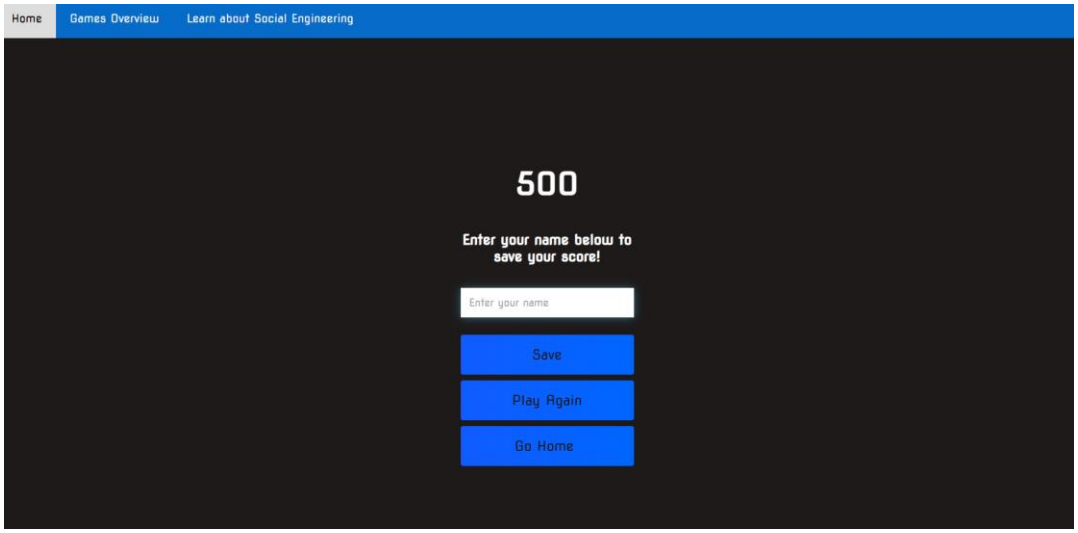


Figure 6 - Results Page

2.2 Page Switching

The page switching functionality allows the user to navigate and easily access multiple pages with a single click per button. This function was used throughout the application, and as such is a key aspect of the DTT. The page switching was made possible using the ‘a href’ code within the HTML pages created within Visual Studio Code. Figure 7 shows an example of the page switching functionality, via a clickable button which is coded to highlight as white when hovered over.

Figure 7 - Page Switching functionality

During the design process, the navigation bar was chosen to be a key component of the tool, as it allows for a consistent place of navigation regardless of which page the user is on. Although each page has slightly differing options within the navigation bar depending on where they are, the consistency of its size, location on the page and colour allows for user familiarity and enhances usability within the tool. Figure 8 shows the mentioned navigation bar, which was a cornerstone of the navigation within the tool.

```
<div class="topnav">
  <a href="NewSocialEngineeringTool.html">Home</a>
  <a href="GamesOverview.html">Games Overview</a>
  <a href="Learn about Social Engineering.html">Learn about Social Engineering</a>
</div>
```

Figure 8 - Navigation Bar HTML Code

During the design and implementation of the page switching functionality, it was realised that there was errors when attempting to navigate to the desired page. This was due to the file locations of the destination page, as it was at times in a different folder or directory from the current page. This was realised and fixed, by using the '..' and '/' keys to assist in navigating to the correct directory within the code. The benefit of this was that the pages were able to redirect to the intended area, but the files were able to be kept in a logical and organised manner, which is beneficial when being shared for viewing at the end of the implementation and testing process. Figure 9 shows the code used to navigate in and out of the relevant folders to maintain an organised view.

```
Click <a href="Learn about Social Engineering.html">'True or False Quiz' to test yourself!</a></p>
```

Figure 9 - Page Navigation Code

Without the functionality mentioned via the page switching functionality, the DTT would now be able to facilitate the seamless and consistent page switching that currently exists within the training tool. In particular, this could have either led to a significant delay in project progress, as a new method of navigation would have to be identified. The lack of this feature could also lead to project failure, as the additional time or lack of options in finding a resolution would mean the user would be unable to navigate between pages. This would then defeat the purpose of the training tool, and significantly impact the user experience and learning benefit. The page switching functionality also facilitated the pages to be separate from each other in their own right, and enables individual development of each page, using the agile methodology whereby each page can be iterated upon. There is also opportunity for the user to navigate to the next learning area through the hyperlink displayed at the bottom of some of the pages. The benefit of this is that it allows the user to navigate naturally through the pages, keeping their learning experience consistent and minimising disruption according to the reviewed literature within the first deliverable (Darejeh, 2013) This is also mentioned further in the GUI section. Hyperlink but also interlinked through the logical flow of navigation shown below in Figure 10.

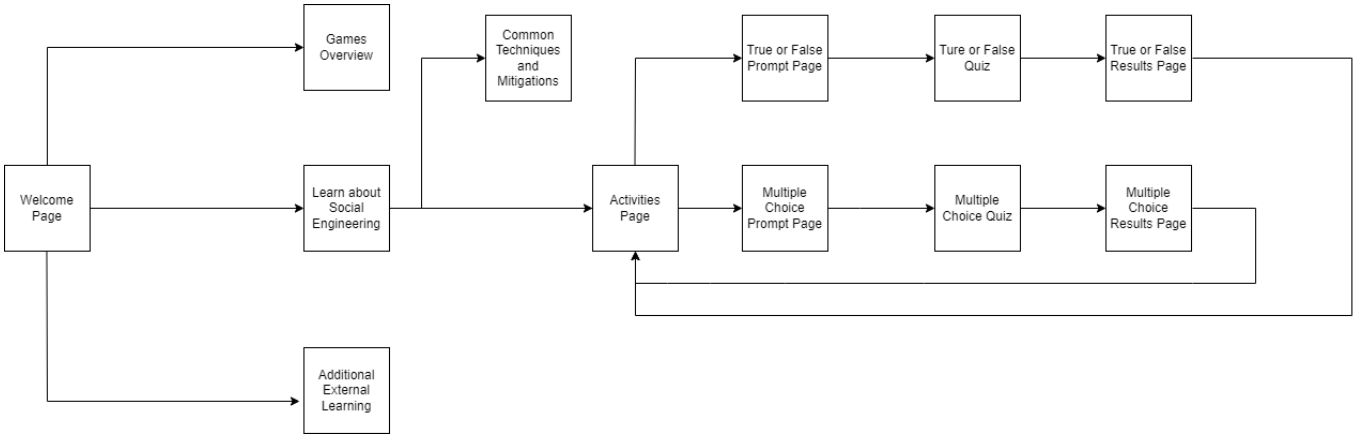


Figure 10 - Logical Flow Overview

Additionally, the performance of the page navigation function is high, as the function is consistent and can be adapted to interlink with any other created page. Due to the smaller content and backend loading in comparison to a mainstream platform provider such as Facebook, the load time for page navigation is one second or less. This allows the user to remain engaged within the tool and reduces the possibility of distractions while the user is waiting for the page to load. The consistent user experience designed by this function, will ensure the user can learn and memorise as much social engineering content as possible from the tool, helping to keep themselves and others safe in the future.

This functionality also caters for the progress bar being set to default within the assessments, which is present to indicate to the user how far they have progressed within the assessments. This provides consistent and easy to read feedback, which encourages the user to continue engaging with the assessment until the bar has been filled. By default, when the assessment pages are loaded, the progress bar is set to 1 out of 10 as that is the initialised default parameter within the JavaScript code. Figure 11 shows the progress bar at the start of the assessment.

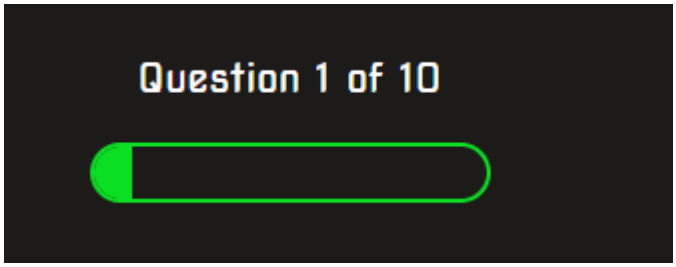


Figure 11 - Progress Bar

2.3 Graphical User Interface

2.3.1 Background Information

The Graphical User Interface (GUI) is a key aspect of the DTT, as it provides the base functionality that all other aspects of the tool are governed by. Components such as the page switching and quizzes use the GUI to provide the consistent appearance of the tool, which allows for efficient functioning and also familiarity for the user, which is shown to boost engagement within a training tool (Alkhamis, E. & Renaud, K. 2016)**.** It is due to this that the GUI can be considered the most significant aspect of the DTT. Additionally, the GUI's key aim is to provide the user with an interface that is simple to use, and also effective at achieving its purpose, which is to navigate the user to the various locations within the tool, such as the learning area and assessments that provide the user with a varied form of learning through the text and images. Consequently, the GUI allows the user to engage with the relevant training material, meaning that they improve their future actions to keep themselves and their peers safe from social engineering attacks.

As the GUI is a key component, there was careful consideration needed regarding the design and implementation to produce a useful and effective training tool. Therefore, it was necessary to be aware of any potential variables or issues that could cause the GUI to fail throughout the user navigation or even simple displaying on the pages. The GUI can also be difficult to produce when the coding is varied between pages, as the quiz pages use JavaScript for the logic functions, whereas the learning area pages only use HTML and CSS to display their content.

As a result of this, the HTML/CSS/JS framework was used to produce the training tool, as various other successful tools have been built using this functionality and programming languages. Due to there being other tools that are commonly built in this format, it provides a level of confidence that this framework could be replicated for this use case.

2.3.2 Buttons

The buttons for the training tool were produced primarily in two locations. The top bar, which is consistent in location, allows for simple navigation within the tool. This was built in two phases, firstly by the HTML code that produces the clickable button. Figure 12 below shows the top bar HTML code.

Figure 12 – Navigation Bar HTML Code

Once the HTML code was written for the button, the CSS code could now be designed and implemented, which produces both the blue styling with white writing, and also the fixed position at the top of the page. The colour choice allowed for a friendlier approach to the tool, and also a sense of clarity in reading what each button is for, as the white text is

contrasted. This was designed to also help support older or vision impaired users in their use of the tool. Figure 13 below shows the code for the top navigation bar CSS styling.

```css
/* Navigation bar styling */
.topnav {
  overflow: hidden;
  background-color: rgb(7, 107, 201);
}

/* Navigation bar links styling */
.topnav a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  font-size: 20px;
  padding: 22px 24px;
  text-decoration: none;
}

/* Change color on hover */
.topnav a:hover {
  background-color: #ddd;
  color: black;
}
```

Figure 13 - Navigation Bar Styling Code

It is useful to note that hyperlinked lines of text were created, and although they were not styled to be presented as buttons, they do offer seamless transition for the user between the pages, particularly those pages within the learning area. Figures 14 and 15 below show both an example of the linked text, and the HTML code that was written to produce this functionality.

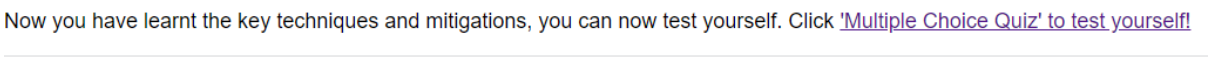


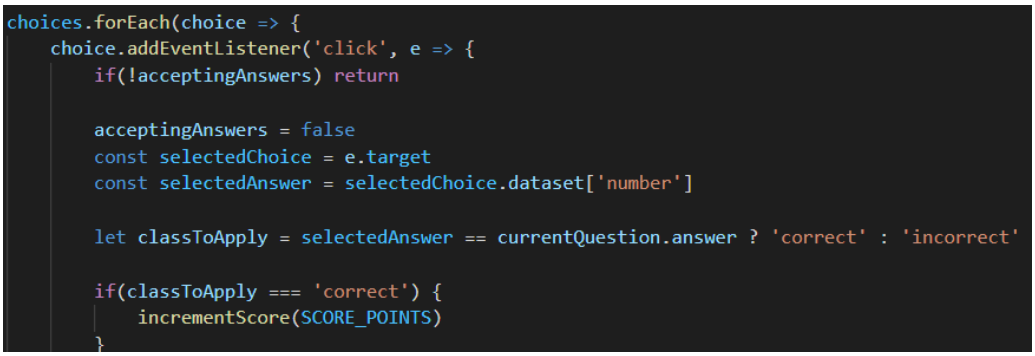
Figure 14 - Example of Hyperlinked Text

```
Click <a href="True or False files/True or False Prompt.html">'True or False Quiz' to test yourself!</a></p>
```

Figure 15 - Hyperlink HTML Code

From this functionality, hyperlinked text that was simple to produce could be placed at the bottom of learning pages, as an alternative for the user having to scroll back to the top of the page to proceed via the navigation bar. This allows the user to remain engaged with the current learning material, and also supports the users who may not be as computer literate when using the tool.

Additionally, buttons were created in the quiz pages, with JavaScript functionality interlinked that allowed for a higher degree of functionality than what HTML and CSS can provide alone. Section 2.3.3 will provide more detail as to how the JavaScript was interlinked with the pages through the HTML src function. These buttons, although act similar in function to the regular buttons, are actually a key component in allowing the user to select the correct answer within both quizzes, as there is not a traditional 'Submit' button present within the tool. Figure 16 below shows the JavaScript code behind the user being able to select an answer.



```
choices.forEach(choice => {
    choice.addEventListener('click', e => {
        if(!acceptingAnswers) return

        acceptingAnswers = false
        const selectedChoice = e.target
        const selectedAnswer = selectedChoice.dataset['number']

        let classToApply = selectedAnswer == currentQuestion.answer ? 'correct' : 'incorrect'

        if(classToApply === 'correct') {
            incrementScore(SCORE_POINTS)
        }
```

Figure 16 - Code for User Answer Selection

This functionality, combined with the CSS styling, allows the tool to have responsive and visually clear feedback, and reduces the amount of clicks necessary to proceed, as the user does not need to click submit for each question they have answered, or even for when they have completed the designed set of questions. This is a considerable element of the user experience, and as such there was a need to ensure the answer selecting process was a streamlined as possible with the tool being usable by a wider demographic.

2.3.3 CSS, HTML and JavaScript Integration

Within the training tool development phase, there was a need to interlink the three aspects of the training tool, so that functionality was both seamless and effective. Visual Studio Code as the chosen software provided the relevant functionality to enable this, as it allowed for the CSS and JavaScript (where necessary) to be interlinked with the HTML page, meaning that all components were loaded together and without clashing. The consideration towards where these scripts were called within each page is significant and was heavily considered, as the location at which these elements were interlinked could have caused significant issues, such as the pages not loading or loading not as intended. Figure 17 and 18 shows the CSS and JavaScript scripts being linked into the HTML page. As a result, correct location of these scripts allows the tool to function as intended, and with the correct styling and layout applied within the CSS scripts for each element of the page.

Figure 17 - CSS Linked into HTML Page

Figure 18 - JavaScript Linked into HTML Page

2.3.4 Score and Current Question Counters

The score counter was created using JavaScript code and was produced by using the designed scoring mechanism of every correct answer being worth 100 points, with a total of 1000 points available in each quiz. If the user gets the answer correct, the counter increases by 100, and of course if incorrect, the current score remains. The research conducted indicates that a scoring system is efficient at sustaining and can also provide a level of competition between users, which can increase and sustain engagement over time, (Erhel, S. & Jamet, E. 2021). The way it was implemented in the tool also means that users are more likely to return to the tool as time progresses, trying to beat their previously set score. Figure 19 below shows the score counter mechanism, written in JavaScript.

```
incrementScore = num => {
    score +=num
    scoreText.innerText = score
}
```

Figure 19 - Score Counter JavaScript Code

The current question counter was created using JavaScript code, and what is essentially a ‘For’ loop**.** This was designed and implemented so that the loop is checking after each new

question is loaded if the current question is greater than 10. Once the user has exceeded the 10 questions asked, the user would be redirected to the results page. Figure 20 shows the JavaScript code behind this functionality.

```
questionCounter++
progressText.innerText = `Question ${questionCounter} of ${MAX_QUESTIONS}`
progressBarFull.style.width = `${(questionCounter/MAX_QUESTIONS) * 100}%`
```

Figure 20 – Current Question Counter JavaScript Code

3. Development Lifecycle

3.1 General Background

The development of the DTT, as discussed previously, required various components to not only be compatible, but work flawlessly and effectively together to produce the desired results, and achieve the previously set project aims and objectives. Choosing and implementing the relevant development methodology was significant in ensuring that the DTT was completed to a high standard, and deadlines for the project were met. Additionally, if the development methodology were found to be incompatible or inefficient for the project, it could result in the DTT being inadequate in terms of the end look and feel or feature set. An incorrectly chosen methodology could also lead to delaying of deadlines, or at worst, project failure as the tool would not be up to the required standard. The significance of this, was factored into choosing the development methodology, and also considering the required features and functionality within the proposed training tool. External factors that could also impact product completion were also considered and noted, as this would ensure that there was not any aspects that could fail that were not already considered prior to development.

Additionally, the agile methodology chosen brings useful benefits for the use case of the training tool. Agile based methodologies are often governed by the agile manifesto, which contain a number of principles and actions that should generally be followed to be considered as such. One such priority is flexibility, to allow changes even those that would be considered late in the current process**.** (Gustavsson, 2016)This allows for swift functional, yet early stage development of features that would then be built out after further iterations to produce the functional end output, such as buttons for page navigation. This allows the development lifecycle to be proactive and not reactive to such unforeseen variables that may affect the project. The swift prototyping of features and functionality also supported external factors surrounding the project, such as when other academic work had to be submitted. Whilst the bulk of the actions were managed through the project plan, the prototyping of features meant that external factors could be managed better, allowing for a greater level of flexibility, in order to continue to meet the agreed objectives.

3.2 Implementation

With the aforementioned agile methodology benefits identified, this was then inputted into the development lifecycle of the DTT to ensure that the development could be both manageable and concentration upon to deliver the end product. From this, a SCRUM based development lifecycle was chosen, as it did not only cover the key principle of the agile methodology, but it also contains a key aspect of the process that suits the type of project that this was. The key aspect being iterative development over time, particularly through sprint. Typically, in sprints, the SCRUM Master will provide the development team with design tasks, to ensure that the implemented functionality can be layered over the top of the design of said function. This would be for the use of buttons, as there is a design aspect to these, as well as the key functionality which could be considered as layered on top. Once these tasks are agreed upon between the SCRUM Master and the client, then they are added into a Sprint Backlog, which is then allocated and prioritised to ensure the seemingly more significant features are implemented first.

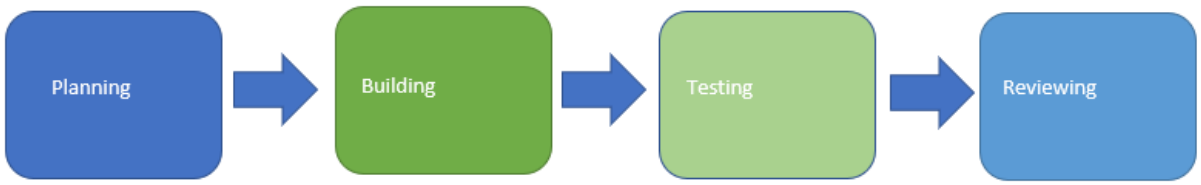


Figure 20 - Sprint Stages

Figure 21 - The Scrum Framework (Darwish, N.R and Megahed, 2016)

With this in mind, it was identified that the corporate, team based style that this framework is dependent on, is not available due to the individual nature of this project. Consequently, Kanban was chosen to be an additional methodology used in conjunction with agile towards the later stages of development, to achieve the objectives associated with this project. A significant feature of Kanban is the ‘Kanban Board’ where all design tasks are identified, and centrally controlled. The board allows the organisation of tasks, in a simple status manner of ‘In Progress’ ‘To Do’ ‘Complete’ and sometimes ‘Backlog’ which would be for items that are in backlog until the next sprint, which usually run on a monthly basis. This gives the developer a top-down view of the current status of each section of development, allowing for prioritised actions, and reduced level of risk.

Furthermore, Kanban has other aspects that make this framework more usable than SCRUM for this particular project. One of them is the casual informality that Kanban provides. As the project is designed to be managed and implemented by one individual, the lack of SCRUM Master means the outstanding tasks can be managed and prioritised individually and internally (Ahmad, Markkula and Oivo, 2013). On the other hand, this may cause project issues and potential failure, as there is a lack of direction from someone such as a SCRUM Master, this could lead to mismanagement and prioritisation of actions within the sprint (Department of Computer Science, Virtual University of Pakistan, and Ashraf, 2017)**.** This is a consideration, although a limited one as the project is undertaken on an individual basis, which makes this framework ideal for this scenario.

3.2.1 Planning

The planning stage, which exists at the beginning of the sprint, is essentially comprised of observing and managing system design documents, such as the GUI overview, Class Diagram and Use Cases which can be found in the 3.2.1 Planning section. These were key documents that provided creation a sense of direction in regard to allocating tasks and objectives, whilst also reducing the overall time taken to build the features within the DTT, as there was a clearer sense of direction. This stage also provided context on how the feature would be tested, to ensure the created outcome was fit for the intended purpose.

From this, the process is such that a feature is chosen from the ‘To Do’ column on the Kanban Board and moved into the ‘In Progress’ column to signify that this feature is ready to be developed on within the ideal start and finish dates provided. When completed, the status column was changed to ‘Complete’ to show that the feature no longer required further development, as seen in Figure 22. As a result, this facilitated clear direction and prioritisation of development tasks, ensuring the project stays on track.

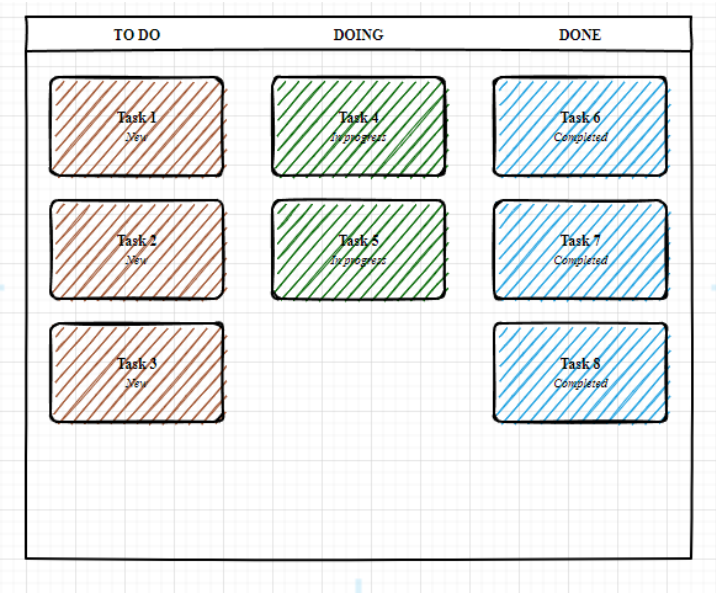


Figure 22 - Kanban Board Example

3.2.2 Building

The building process used a variety of programming aspects as mentioned in the reflection and critical analysis section. This involved creating and programming the features and as such the necessary functionality of the DTT using the relevant design documentation such as the Use Cases, as shown in Figure 23 and Appendix A. Primarily, the GUI of the DTT was created and designed using the Visual Studio Code software. This text edit allowed the HTML elements to be built and assigned variable names for later programming. From there, the relevant CSS and JavaScript pages could be built separately, and then interlinked through the HTML page, allowing the file to be stored in separate locations but work in conjunction with each other. The final stage of the development process is the testing section below.

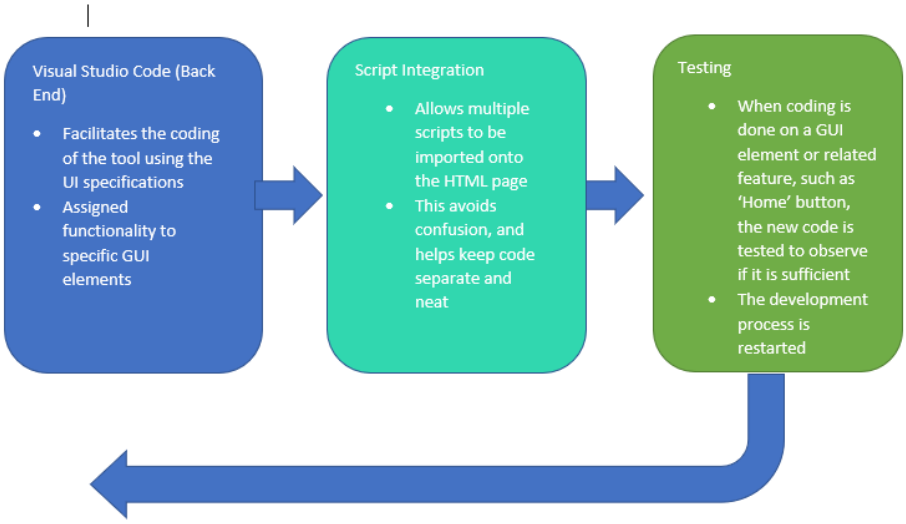


Figure 23 - Building Process

3.2.3 Testing

The testing process in regard to the development lifecycle is extremely crucial in order to deliver an effective training tool that achieves its aims and objectives. Consequently, execution of the correct testing methodology must be thoughtfully chosen as incorrect or false implementation could result in a poor quality training tool. Consequently, it was decided that an Agile testing methodology should be designed and executed to work coordinated with the mentioned development methodology.

The execution of this testing related to the creation of a testing plan which was drafted during the first deliverable. This was then adjusted and improved upon, to accommodate for the changes made during the development lifecycle, such as pages within the DTT being renamed or added/removed. The testing plan which was completed and was used to meticulously test the produced tool to make certain that it will be usable in the future, with the expected and actual outcomes recorded within the tables. The testing plan can be found in Appendix A.

3.2.4 Reviewing

The reviewing process essentially used the results from the testing stage as a guideline to judge the completeness of the created training tool. As such, there was the flexibility to revisit features that may not have worked as expected during the testing stage. This provided essential feedback from developer to developer, as there was time saved in regard to finding user to test, as would be the case with traditional user testing. This allowed features to be adjusted as necessary in a fluid manner.

4. Reflection and Critical Analysis

4.1 Strengths

The DTT application has provided the easy to use platform for users of all ages and abilities to engage with, with the hope of assisting general user groups in improving their future cyber actions. This allowed the development project to achieve the previously set aims and objectives of providing an effective training tool. The success of the project can be considered through the learning area, with a range of pages decided to guide the user through in a friendly format, along with the results pages which have been implemented in conjunction with the wider tool. The finished design of these pages were influenced by the original design documented in the first deliverable, as well as the research conducted in regard to the existing educational digital tools, and wider research on significant aspects such as the user interface. With these considerations implemented into the tool, the user will be able to gain significant knowledge, thus achieving the aims and objectives of the development project.

Another success of the project was the Agile Development Lifecycle that was key in supporting the iterative development of the training tool. In particular, the project delivered a tool that was produced under time constraints, potential issues during development, and external factors. It should also be noted that there was limited exposure to the agile

methodology prior to this development piece, and as such there was elements of ‘learning on the go’ particularly in regard to development process and also the programming languages used. From this, beneficial experience was gained which will only serve to produce improved pieces of software in the future.

An additional success of the projects was learning and implementing the GUI, as there was also a lack of exposure to producing these as a foundation for a functional website. The knowledge gained in this area will also help improve future design considerations, particularly if there was an additional developer who was willing to progress this development project further in the future. This means that the project could also be continued in the long term, whilst also achieving the aims and objectives discussed in the short term for the development project.

Overall, the development project can be considered a success, as a social engineering training tool was delivered that met the aims and objectives of the project. Furthermore, key skills were learnt and developed throughout the development lifecycle, which will support future actions in Cyber Security. The tool will also be able to help users in improving their future actions, which could have a significantly positive impact on the user’s future decisions.

4.2 Challenges

There were a few challenges during the project, however there was possibly less issues than expected at the start of the project. These challenges will be considered when a new project is started, as these lessons learnt may help future actions.

Another challenge was time management during the project. Although the agile methodology allowed for a clear plan, due to external factors, some actions in relation to the project were done in quick succession. Ideally in normal conditions there would be additional team members or resources available to help produce the features quicker, however the individual nature of the project meant that this had to be done alone. Despite this, the project was still delivered successfully, and skills regarding time management were improved upon, which can only serve as a benefit for future projects undertaken.

The most significant challenge during the development project was the learning of the relevant programming languages to produce a tool that is not only user friendly, but also has a level of logic behind it that makes it dynamic. Prior to development, there was no previous experience of HTML, CSS Or JavaScript, meaning that these languages had to be learnt during the development process, by the use of a paid training course. Fortunately, there was personal finance available for this, as without it, there may have been a delayed learning process taking place, which could have impacted initial prototypes and the meeting of objectives, particularly around the first deliverable stage. Learning these skills will also mean that these can be taken into further projects or job roles and is an enhancement of the current skill set of the developer, which creates a level of gratitude towards the process.

4.3 In Hindsight

If the development project were to be attempted again, and in theory the development in question was still a DTT, the hope is the project and objectives would be altered, to accommodate for the lessons learnt. With improved programming language, the prototype features and working functionality would be built out much quicker, meaning there would be

extra time for additional features such as a third quiz. This could, in theory, produce an improved overall product, as the experience gained could be used to produce an increasingly efficient training tool.

Another change that would be taken with hindsight would be an increased level of research in regard to current social engineering training tools, and additional literature would also be found in this area. This is due to the current literature being used as a building block to further implement additional features to enhance the training tool, as more literature would help support decisions made throughout the project and provide additional benefit to the end user.

Another change that would be considered is user testing. As the developer was the only individual who tested the tool, and as such alternate perspectives may provide useful insights as to what can be altered to produce a better tool. It would also help with time management, as users would be able to be the primary tester, instead of the developer.

Finally, in hindsight, the Spot The Difference quiz and learning recommendations would have been implemented. Due to external time constraints and the added learning of the relevant programming languages during development, there was not enough time to effectively implement the third activity along with the learning recommendations based off the users score. This would have been good to add in as added functionality within the tool, however this was not possible. This acts as a lesson learnt for the future, as partly this may have been cause due to being too ambitious considering time available and the coding learning, and also a lesson to improve project and time management. This outcome will ensure that future decisions are considered with these elements in mind. This is also reflected in Appendix A, in the Testing section as some of the results are shown as N/A.

4.4 Software Tools

4.4.1 Visual Studio Code

Visual Studio Code is a source code editor that is available on most operating systems. This software has a range of features that makes web development simpler, such as built in debugging, syntax highlighting, snippets and more. The syntax highlighting and debugging in particular were extremely useful, as the software would scan the code as it is being written for syntax or logical errors that may impact the output code. This reduces the amount of debugging required when loading the web pages, as the functionality was able to identify a lot of these, reducing debugging time and increasing efficiency whilst also keeping the development agile. Visual Studio also allows for the pages to be interlinked, and as previously mentioned this significantly prevented the amount of code on one page within the editor. This was a key aspect of the application, as it allowed each page to be viewed separately, avoiding confusion, and decreasing the time taken to implement features. The time taken to code the pages without this would have been extremely high and incredibly confusing to manage. Figure 24 shows the Visual Studio application.

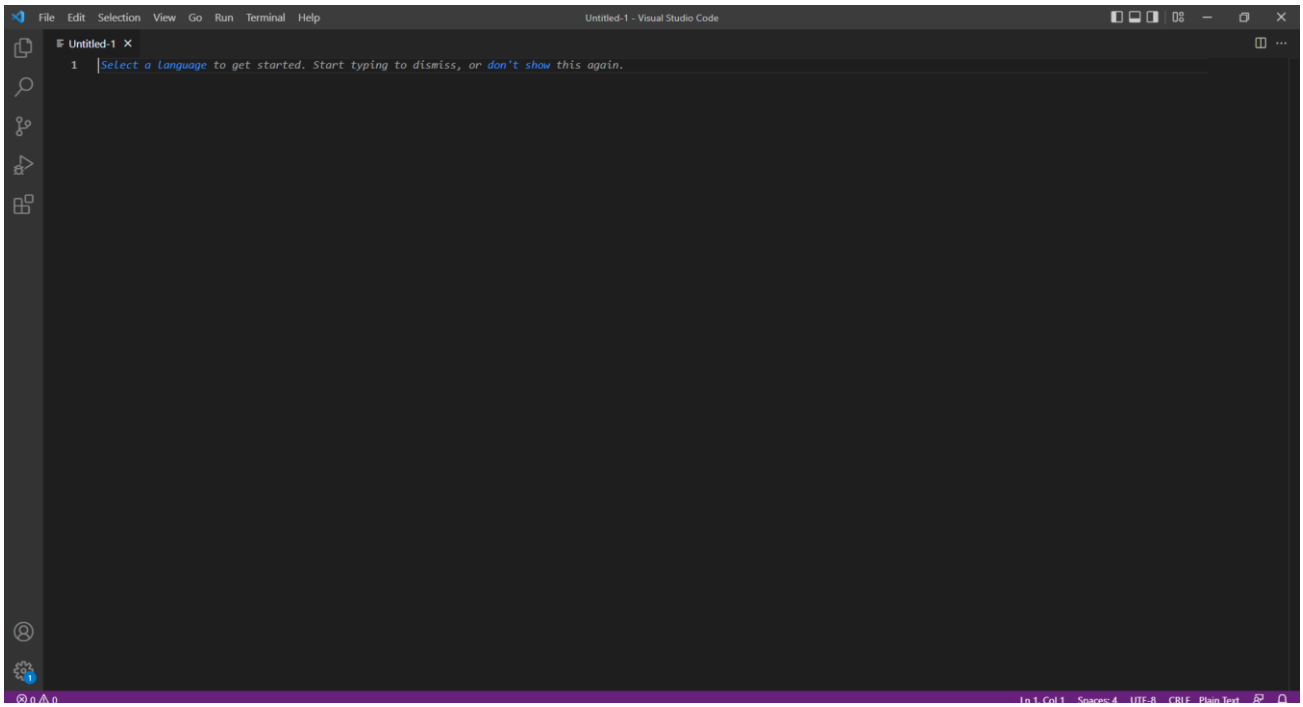


Figure 24 - Visual Studio Code

5. Conclusion

In conclusion, this development project produced a social engineering training tool that meets the aims and objectives of the project. The training tool was built using Visual Studio code, along with HTML, CSS And JavaScript that provided the logical functionality required, which was a key aspect in producing the tool. The user experience has been created through the learning pages, quiz pages and results pages that provide direct and easy to absorb feedback for users in relation to social engineering attacks. Along with this, a software development lifecycle was conducted to manage the aims and objectives of the tool and ensure these were met on time.

Word Count: 7195

6. References

1. Krombholz, K., Hobel, H., Huber, M. and Weippl, E., (2015). Advanced social engineering attacks. *Journal of Information Security and applications*, *22*, pp.113-122.
2. Salahdine, F. and Kaabouch, N., (2019). Social engineering attacks: A survey. *Future Internet*, *11*(4), p.89.
3. Dincelli, E. and Chengalur-Smith, I., (2020). Choose your own training adventure: designing a gamified SETA artefact for improving information security and privacy through interactive storytelling. *European Journal of Information Systems*, 29(6), pp.669-687.
4. Pirocca, S., Allodi, L. and Zannone, N., (2020) December. A Toolkit for Security Awareness Training Against Targeted Phishing. In *International Conference on Information Systems Security* (pp. 137-159). Springer, Cham.
5. Darwish, N.R. and Megahed, S., (2016). Requirements engineering in scrum framework. *International Journal of Computer Applications*, *149*(8), pp.24-29.
6. Ahmad, M.O., Markkula, J. and Oivo, M. (2013) *Kanban in Software Development: A Systematic Literature Review*, p. 16. doi:10.1109/SEAA.2013.28.
7. Darejeh (2013) 'A review on user interface design principles to increase software usability for users with less computer literacy', journal of computer science, 9(11), pp. 1443–1450.
8. Alkhamis, E. & Renaud, K. (2016). The design and evaluation of an interactive social engineering training programme. pp.125-134.
9. Erhel, S. & Jamet, E. (2021) Digital game-based learning: Impact of instructions and feedback on motivation and learning effectiveness [Internet].
10. Gustavsson, T. (2016) 'Benefits of agile project management in a non-software development context – a literature review', p. 11.
11. Department of Computer Science, Virtual University of Pakistan and Ashraf, S. (2017) 'Scrum: An Improved Scrum Process Model', *International Journal of Modern Education and Computer Science*, 9(8), pp. 16–24. doi:10.5815/ijmecs.2017.08.03.

The below section of references are in relation to any images or content used throughout the tool:

12. https://us.norton.com/internetsecurity-emerging-threats-what-is-social-engineering.html for the facts on 'learn about social engineering and 'Common techniques page'
13. Learn about Social Engineering attack lifecycle image - https://lzomedia.com/blog/common-social-engineering-attack-strategies/
14. Common techniques and mitigations Phishing attack image - https://www.cloudflare.com/en-gb/learning/access-management/phishing-attack/
15. Common techniques and mitigations Baiting attack image - https://techviral.net/common-social-engineering-attack-tactics-and-how-to-prevent-them/
16. Common techniques and mitigations Tailgating attack image- https://markets.businessinsider.com/news/stocks/openpath-solves-a-massive-security-problem-facing-businesses-tailgating-1028466597
17. Common techniques and mitigations Pretexting attack image- https://us.norton.com/internetsecurity-online-scams-what-is-pretexting.html

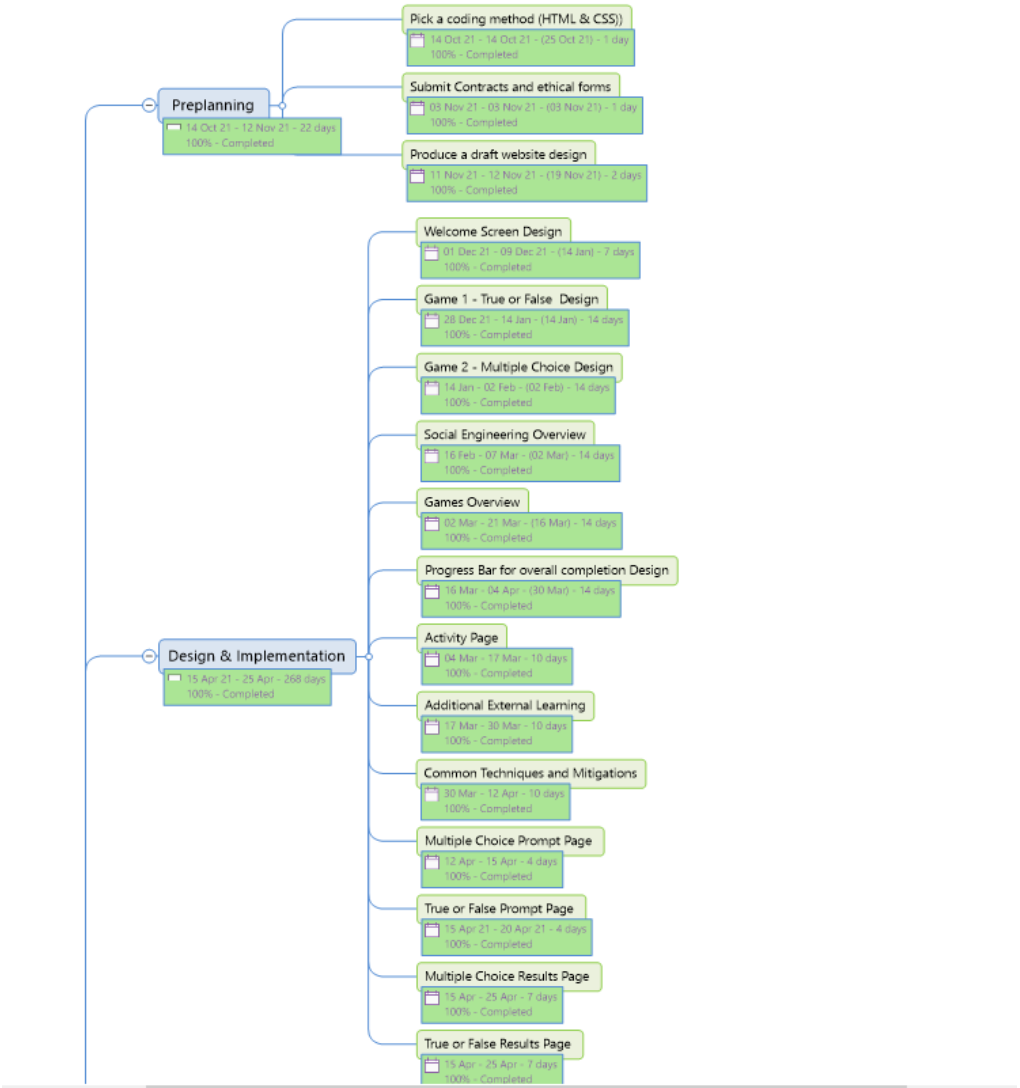
18. Common techniques and mitigations pretexting image **- https://us.norton.com/internetsecurity-online-scams-what-is-pretexting.html**
19. Info paraphrased from 'Common techniques page' - https://www.tripwire.com/state-of-security/security-awareness/5-social-engineering-attacks-to-watch-out-for/
20. Common techniques and mitigations Scareware image - https://sectigostore.com/blog/what-is-scareware-and-scareware-examples/
21. Additional External Learning – left column - https://www.itgovernance.co.uk/social-engineering-attacks
22. Additional External Learning – middle column https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=social+engineering&btnG=&oq=social
23. Additional External Learning – right column https://portswigger.net/daily-swig/social-engineering
24. Common techniques and mitigations Phishing mitigations info (Common techniques and mitigations) https://www.graphus.ai/blog/how-to-identify-and-mitigate-phishing-attacks/
25. Common techniques and mitigations Baiting mitigations info (Common techniques and mitigations) https://www.passcamp.com/blog/what-is-a-baiting-attack-and-how-to-prevent-it/
26. Common techniques and mitigations Baiting mitigations info (Common techniques and mitigations)
27. Common techniques and mitigations Tailgating mitigations info (Common techniques and mitigations) https://blog.dormakaba.com/4-ways-to-prevent-tailgating/
28. Common techniques and mitigations pretexting mitigations info https://www.csoonline.com/article/3546299/what-is-pretexting-definition-examples-and-prevention.html
29. Common techniques and mitigations Scareware mitigations info (Common techniques and mitigations) https://clario.co/blog/recognize-prevent-scareware/#:~:text=Here's%20what%20you%20can%20and,suspicious%20subjects%20from%20unknown%20senders

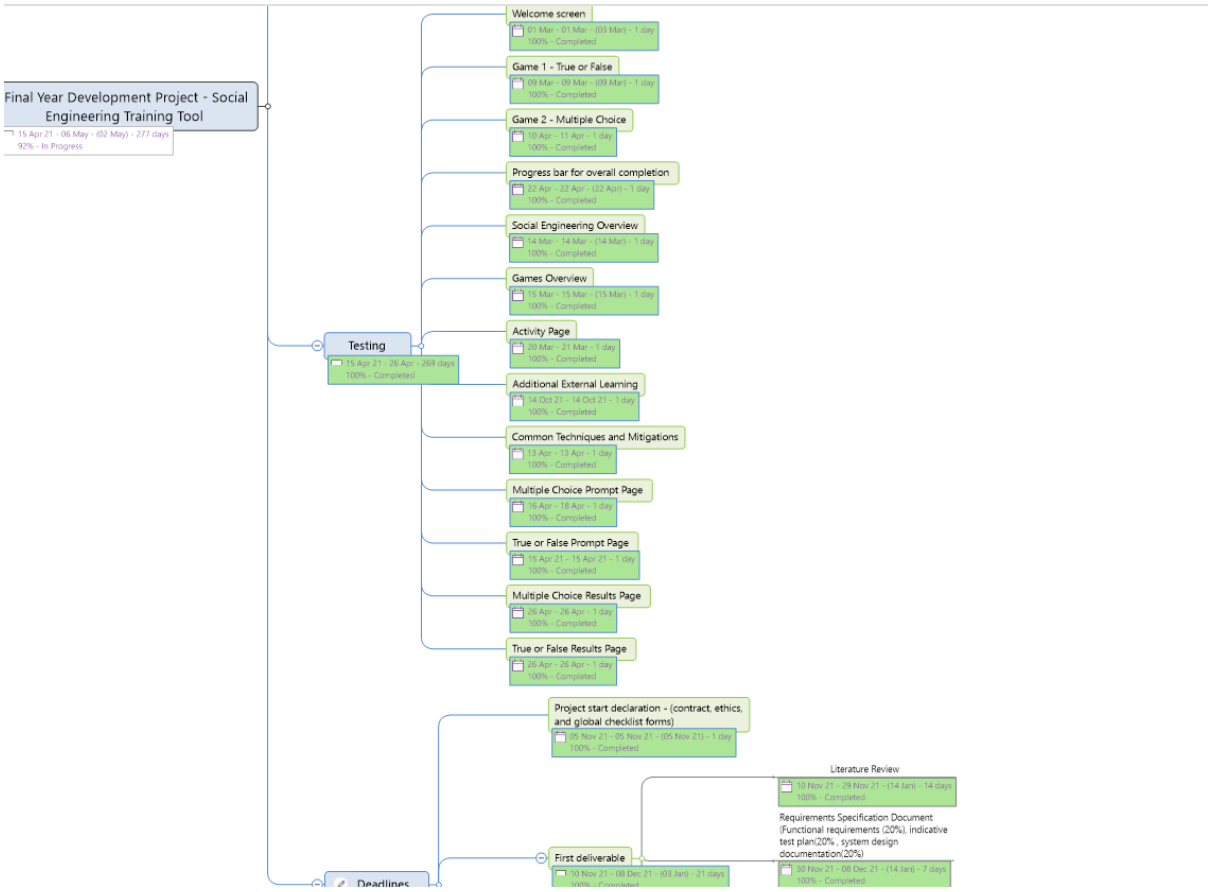
7. Appendices

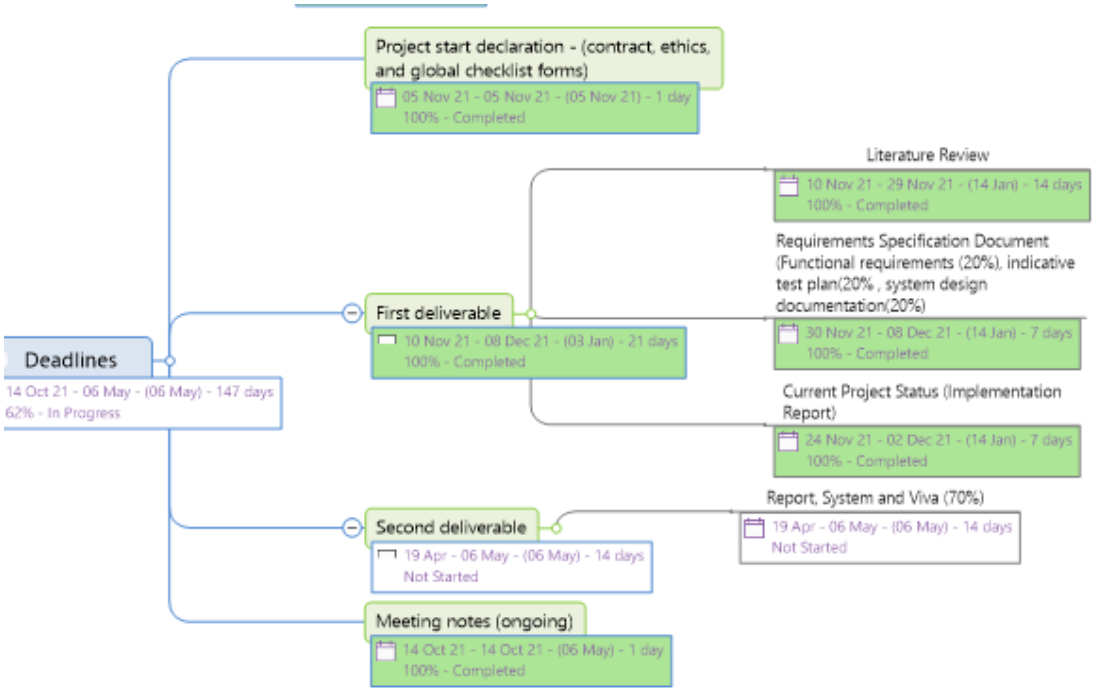
Appendix A

Planning

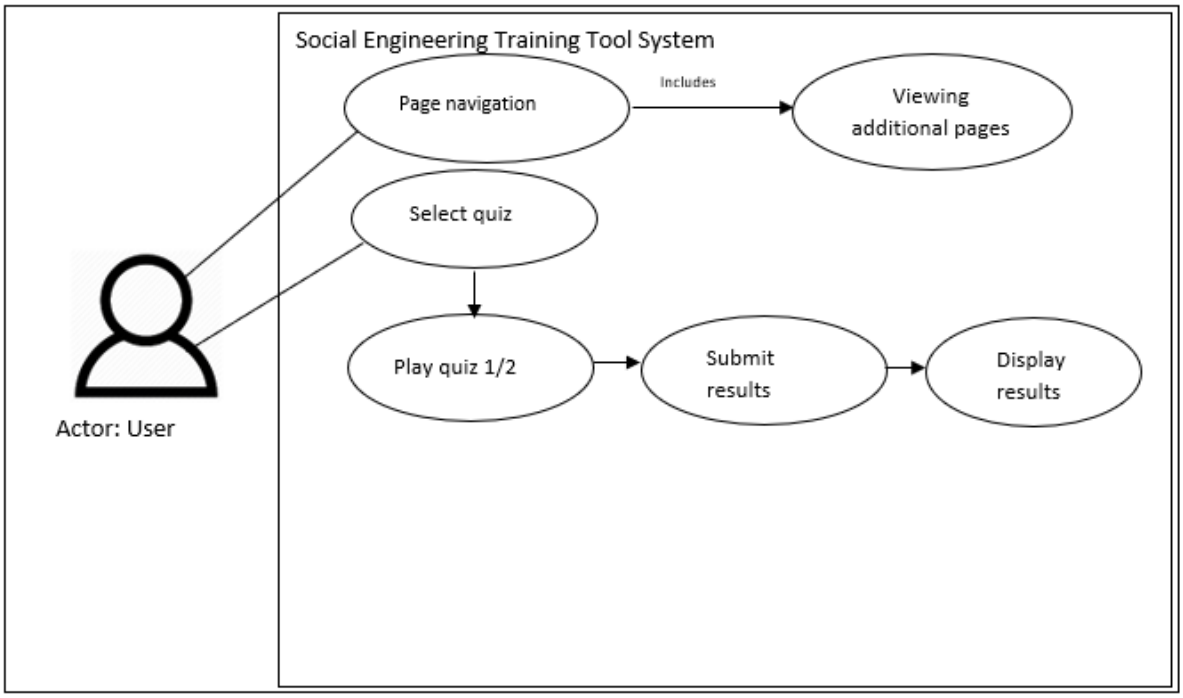
Overall Project Plan:




Final Year Development Project - Social Engineering Training Tool
15 Apr 21 - 06 May - (02 May) - 277 days
92% - In Progress
Welcome screen
01 Mar - 01 Mar - (03 Mar) - 1 day
100% - Completed
Game 1 - True or False
09 Mar - 09 Mar - (09 Mar) - 1 day
100% - Completed
Game 2 - Multiple Choice
10 Apr - 11 Apr - 1 day
100% - Completed
Progress bar for overall completion
22 Apr - 22 Apr - (22 Apr) - 1 day
100% - Completed
Social Engineering Overview
14 Mar - 14 Mar - (14 Mar) - 1 day
100% - Completed
Games Overview
15 Mar - 15 Mar - (15 Mar) - 1 day
100% - Completed
Testing
15 Apr 21 - 26 Apr - 269 days
100% - Completed
Activity Page
20 Mar - 21 Mar - 1 day
100% - Completed
Additional External Learning
14 Oct 21 - 14 Oct 21 - 1 day
100% - Completed
Common Techniques and Mitigations
13 Apr - 13 Apr - 1 day
100% - Completed
Multiple Choice Prompt Page
16 Apr - 16 Apr - 1 day
100% - Completed
True or False Prompt Page
15 Apr 21 - 15 Apr 21 - 1 day
100% - Completed
Multiple Choice Results Page
26 Apr - 26 Apr - 1 day
100% - Completed
True or False Results Page
26 Apr - 26 Apr - 1 day
100% - Completed
Project start declaration - (contract, ethics, and global checklist forms)
05 Nov 21 - 05 Nov 21 - (05 Nov 21) - 1 day
100% - Completed
Literature Review
10 Nov 21 - 29 Nov 21 - (14 Jan) - 14 days
100% - Completed
Requirements Specification Document (Functional requirements (20%), indicative test plan(20% , system design documentation(20%)
First deliverable
10 Nov 21 - 08 Dec 21 - (03 Jan) - 21 days
30 Nov 21 - 08 Dec 21 - (14 Jan) - 7 days
100% - Completed
Deadlines


Project start declaration - (contract, ethics, and global checklist forms)
05 Nov 21 - 05 Nov 21 - (05 Nov 21) - 1 day
100% - Completed
Literature Review
10 Nov 21 - 29 Nov 21 - (14 Jan) - 14 days
100% - Completed
Requirements Specification Document (Functional requirements (20%), indicative test plan(20% , system design documentation(20%)
First deliverable
10 Nov 21 - 08 Dec 21 - (03 Jan) - 21 days
100% - Completed
30 Nov 21 - 08 Dec 21 - (14 Jan) - 7 days
100% - Completed
Current Project Status (Implementation Report)
24 Nov 21 - 02 Dec 21 - (14 Jan) - 7 days
100% - Completed
Deadlines
14 Oct 21 - 06 May - (06 May) - 147 days
62% - In Progress
Report, System and Viva (70%)
Second deliverable
19 Apr - 06 May - (06 May) - 14 days
Not Started
19 Apr - 06 May - (06 May) - 14 days
Not Started
Meeting notes (ongoing)
14 Oct 21 - 14 Oct 21 - (06 May) - 1 day
100% - Completed

Use Case Diagram:



Use Cases:

ID:	SYS001
Title:	Selecting Quiz
Description:	Three clickable buttons that when clicked upon, will then begin the relevant game.
Primary Actor:	User
Preconditions:	Website must be opened.
Postconditions:	The relevant quiz is displayed
Success Scenario:	The user can select from the available buttons for the game they would like to play, and once done this will load the game
Extensions:	If there is a programming related error, the game is unlikely to be displayed when the user interacts with the menu buttons
Frequency of Use:	Low
Status:	Working
Owner:	Michael Green
Priority:	High

ID:	SYS002
Title:	Page Navigation

Description:	The ability for the user to navigate the website, including viewing additional educational pages
Primary Actor:	User
Preconditions:	Website must be opened; buttons must be working.
Postconditions:	The user can successfully navigate the tool
Success Scenario:	Any interactivity within the website design can be used simply and effectively by the user.
Extensions:	N/A
Frequency of Use:	High
Status:	Working
Owner:	Michael Green
Priority:	High

ID:	**SYS003**
Title:	Play Quiz
Description:	Landing on the quiz page will allow each quiz to start
Primary Actor:	User
Preconditions:	Website must be opened; user must be on one of the quizzes
Postconditions:	The quiz will start
Success Scenario:	The user can select the quiz which will then load the quiz
Extensions:	N/A
Frequency of Use:	Medium
Status:	Working
Owner:	Michael Green
Priority:	High

ID:	**SYS004**
Title:	Answer Questions
Description:	The user will be able to select a single answer or a selection of answers.
Primary Actor:	User
Preconditions:	Website must be opened; user must be on one of the quizzes.
Postconditions:	The user has selected their answers
Success Scenario:	The user has selected the answers and is ready to click the submit button.
Extensions:	N/A
Frequency of Use:	Medium
Status:	Working
Owner:	Michael Green
Priority:	High

ID:	**SYS005**
Title:	Submit Results

Description:	The user can submit their results and the view results page becomes accessible.
Primary Actor:	User
Preconditions:	The user has completed the quiz
Postconditions:	The user is ready to submit
Success Scenario:	The user can submit the results without errors and is taken to the results page
Extensions:	Message box is displayed if there is an error such as the user not giving an answer on a specific question
Frequency of Use:	Medium
Status:	Working
Owner:	Michael Green
Priority:	High

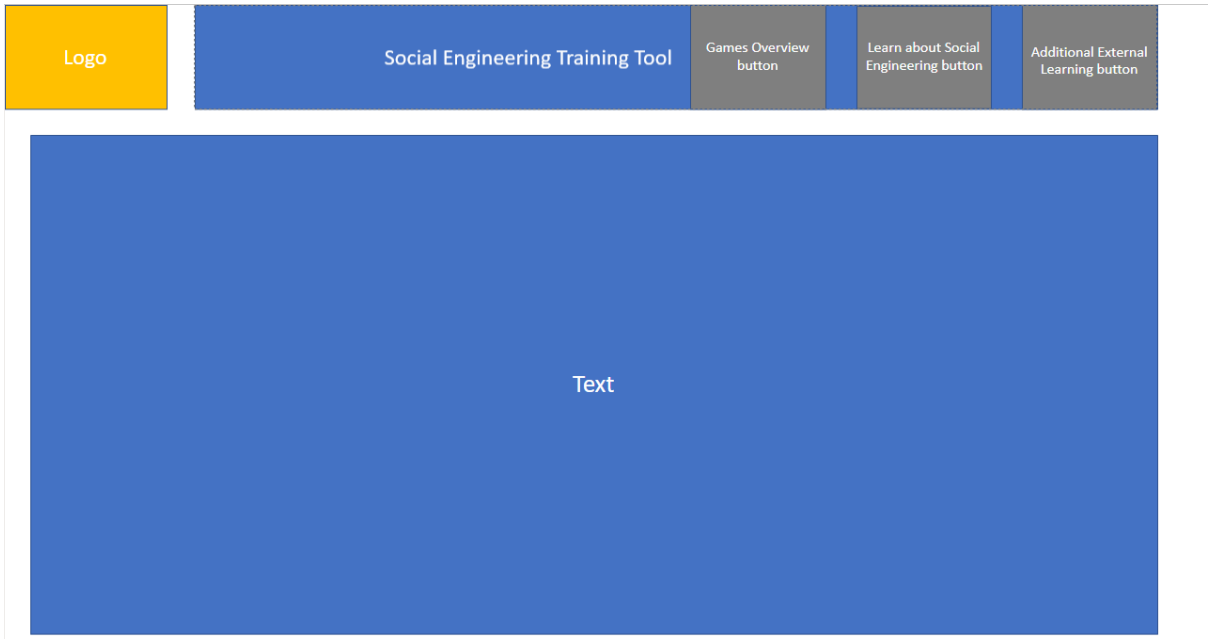
ID:	**SYS006**
Title:	Display Results
Description:	The user, after submitting their results on a particular quiz, is displayed with the results page, and learning recommendations.
Primary Actor:	User
Preconditions:	The user has submitted the relevant answers using the ‘Submit’ button.
Postconditions:	The results page is displayed with the correct information regarding the selected answers.
Success Scenario:	The assessments results page is displayed with the relevant learning recommendations within the same page.
Extensions:	N/A
Frequency of Use:	Medium
Status:	Working
Owner:	Michael Green
Priority:	High

ID:	**SYS007**
Title:	View Additional Pages
Description:	The user can view additional pages provided to support their learning before they are quizzed.
Primary Actor:	User
Preconditions:	The user has navigated to the relevant learning pages.
Postconditions:	The user can view the relevant learning pages.
Success Scenario:	The user can submit the results without errors and is taken to the results page.
Extensions:	N/A
Frequency of Use:	Medium
Status:	Working
Owner:	Michael Green

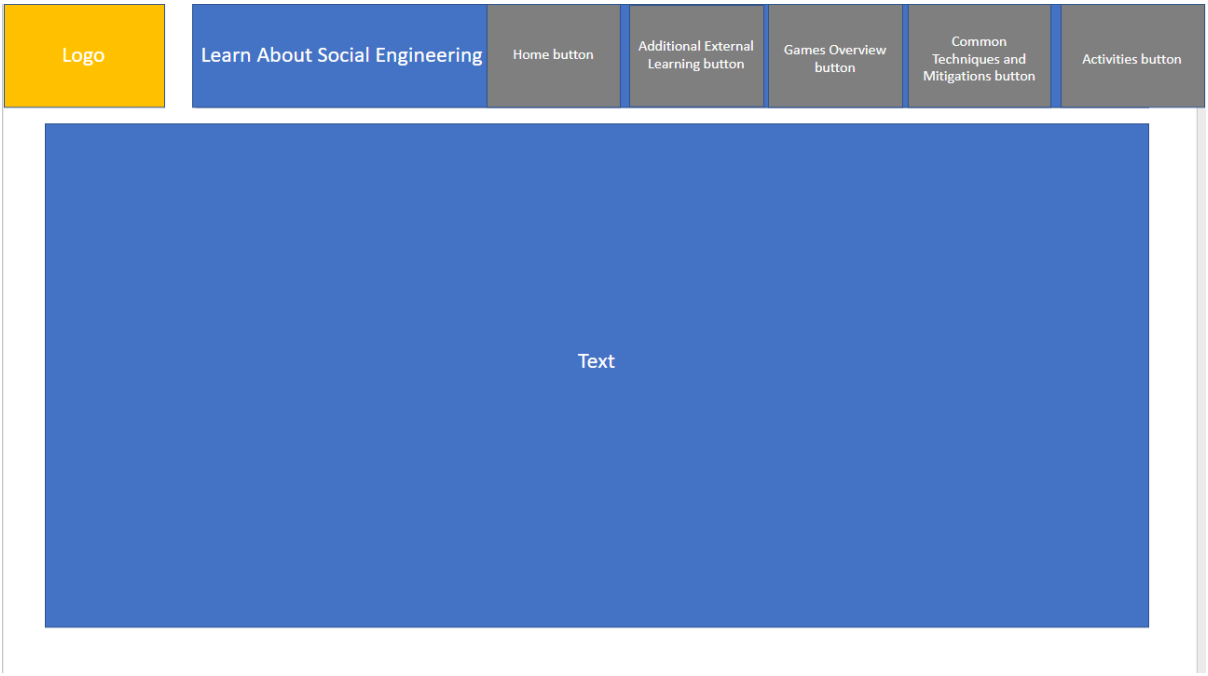
Priority:	High

GUI Designs:

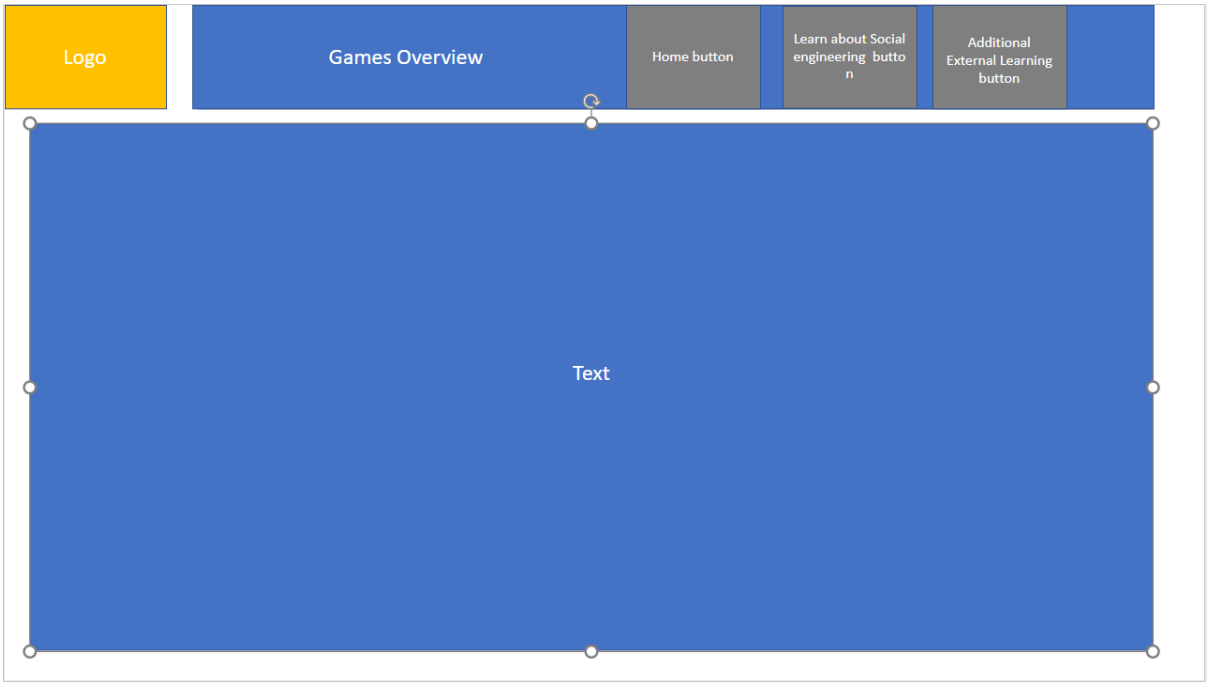
Application UI – Welcome Page



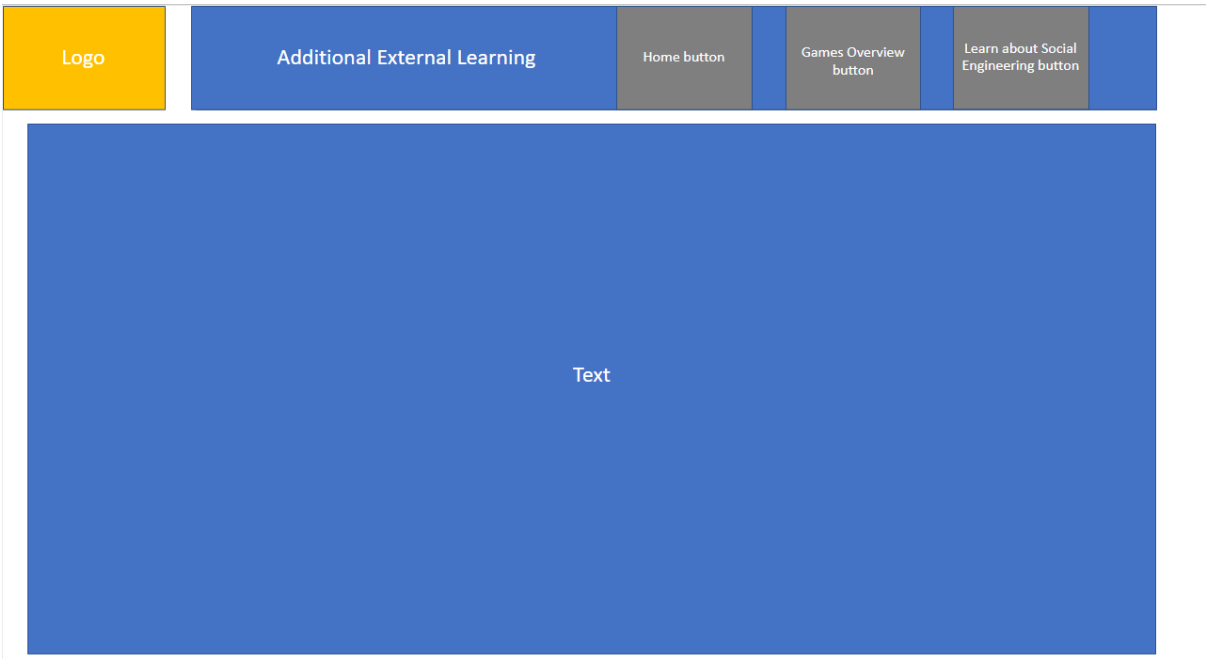
Application UI – Learn about Social Engineering Page



Application UI – Games Overview Page

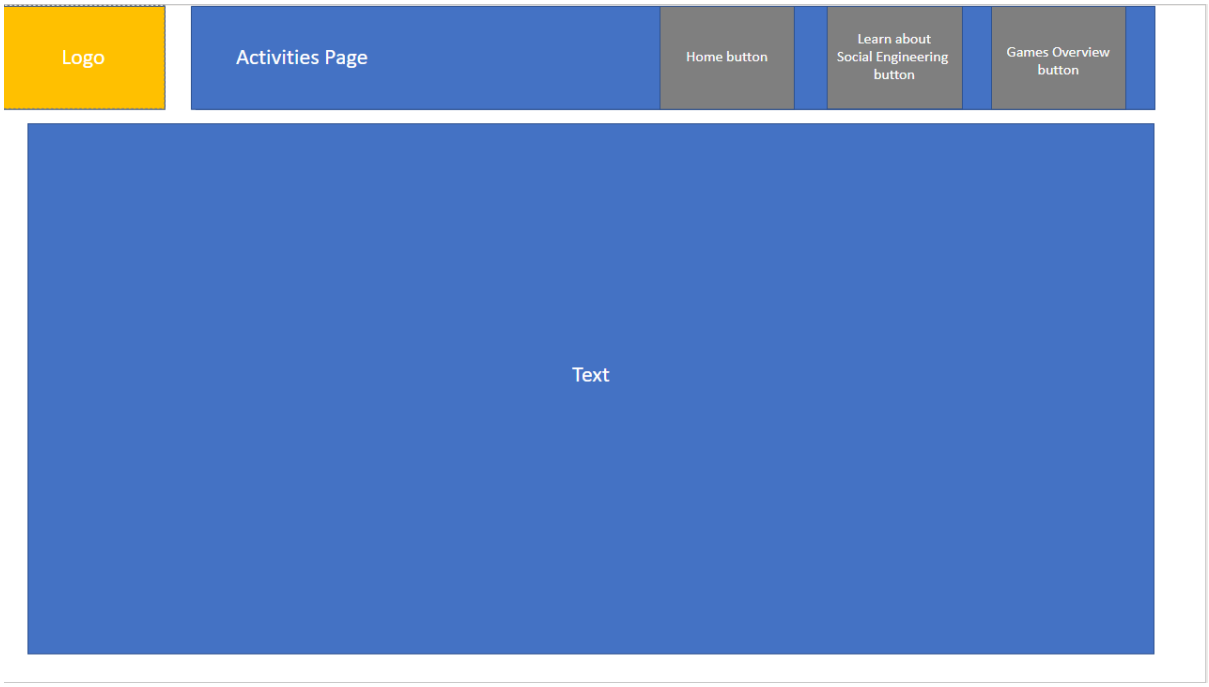


Application UI – Additional External Learning Page

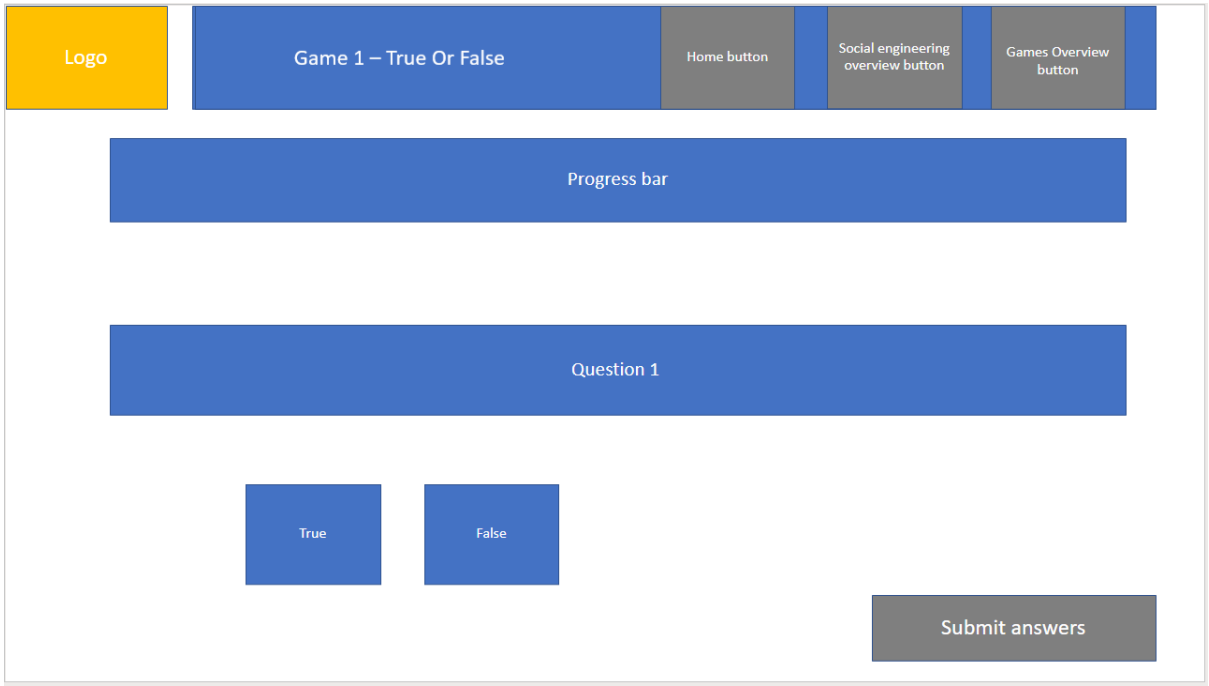


Application UI – Common Techniques and Mitigations Page

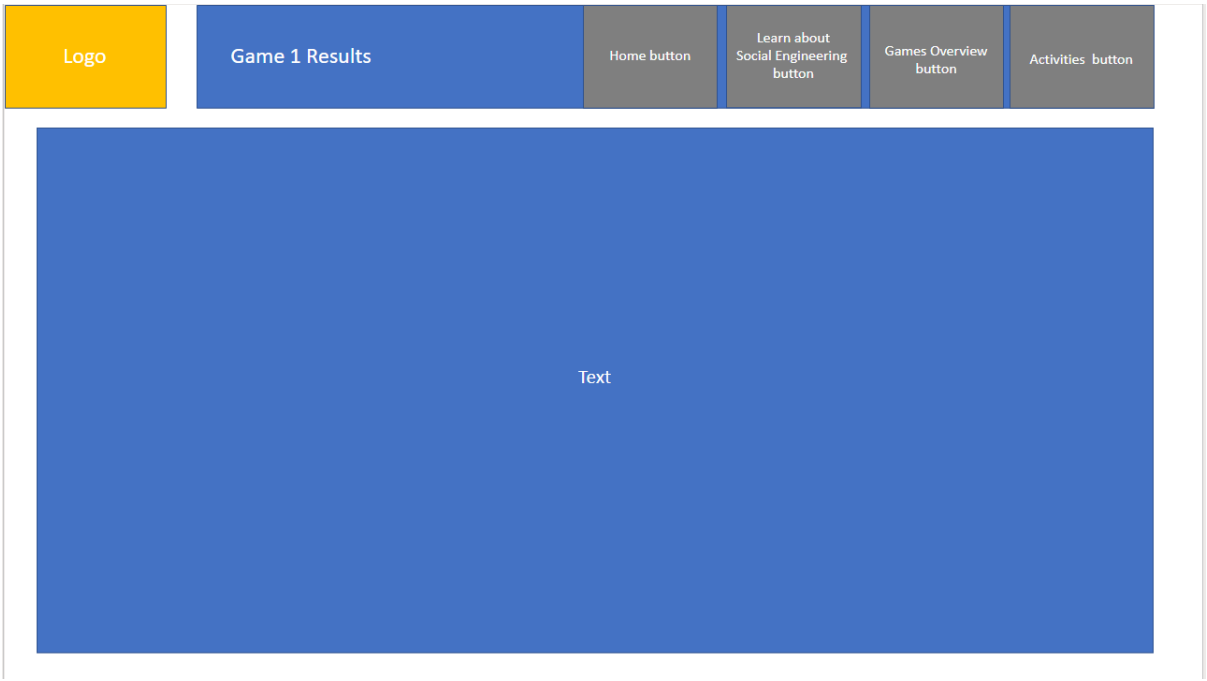
Application UI – Activities Page



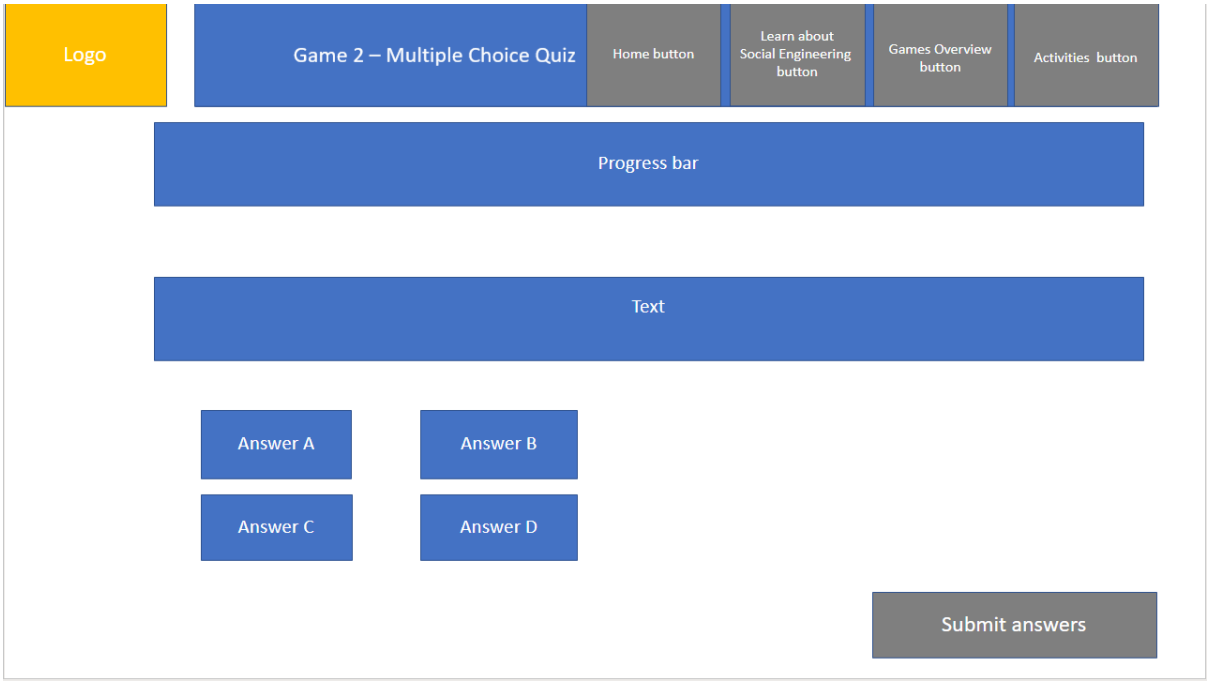
Application UI – Game 1 Page



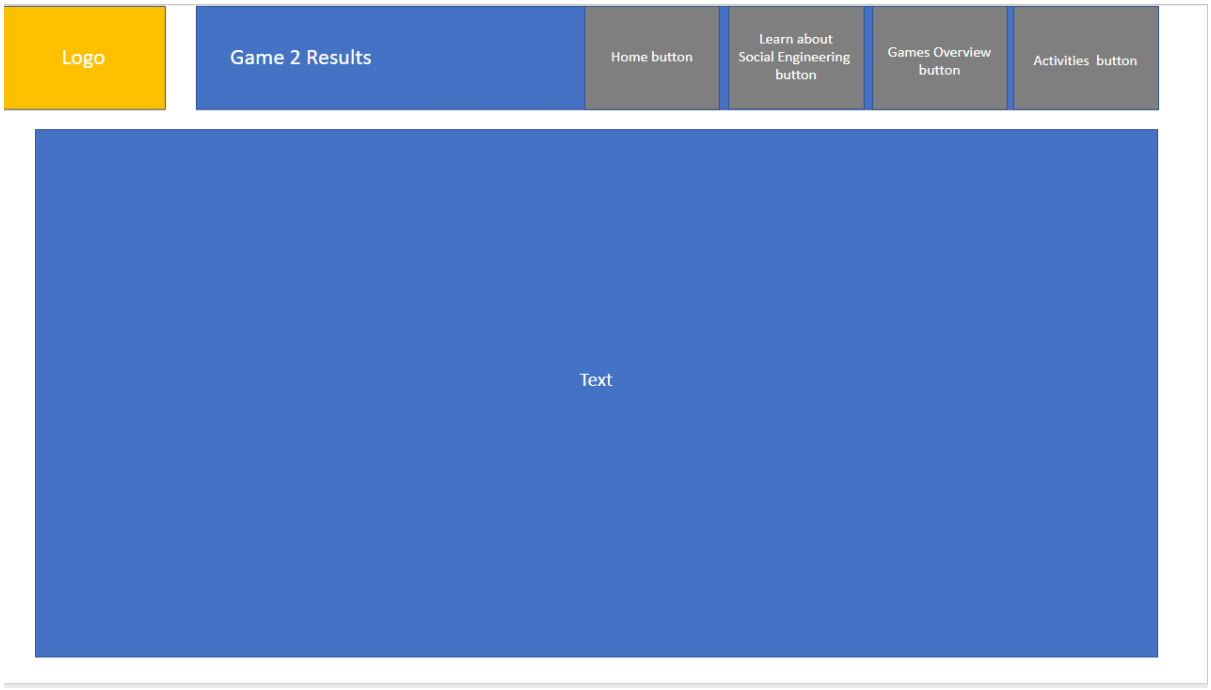
Application UI – Game 1 Results Page



Application UI – Game 2 Page



Application UI – Game 2 Results Page



Testing

Test Case Results:

Test Case ID	Test Scenario	Prerequisite	Test Steps	Expected Result	Actual Result	Pass/Fail	Comments
001	Test that the website can be opened, and the welcome page is loaded.	The software and files are available. The main html file is opened in a compatible web browser such as Google Chrome.	Open the 'Social Engineering Training Tool' HTML file.	Website homepage is displayed correctly with all expected features and functionality.	Website displays with no error.	Pass	Website opens under 2 seconds.
002	Test that the menu buttons when clicked display the correct page.	Website file is open and running.	Click the menu buttons located within the website header.	Each button navigates to the correct page.	Both buttons display the correct pages.	Pass	Page loads correctly.
003	Verify that the Learn about Social Engineering is accessible.	The website is open, and the user has selected the 'Learn about Social Engineering' button.	Click the 'Learn about Social Engineering'' button.	The page is displayed.	Results display correctly.	Pass	Page loads correctly and is accessible.
004	Verify that the Games overview is accessible.	The website is open, and the user has selected the 'Games Overview' button.	Click the 'Games Overview' button.	The Games overview page is displayed.	The games overview loads and is accessible.	Pass	Page loads correctly and is accessible.
005	Verify that the Additional External Learning page is accessible.	The website is open, and the user has selected the 'Additional External Learning' button.	Click the 'Additional External Learning' button.	The Additional External Learning page is displayed.	The additional external learning loads and is accessible.	Pass	Page loads correctly and is accessible.
006	Verify that the Common Techniques and Mitigations page is accessible.	The website is open, and the user has selected the 'Common Techniques and Mitigations page' button.	Click the 'Common Techniques and Mitigations' page button.	The Common Techniques and Mitigations page is displayed.	The additional external learning loads and is accessible.	Pass	Page loads correctly and is accessible.

007	Verify that the activities page is accessible.	The user is on the 'Learn about Social Engineering' page.	Click the 'Activities' page button.	The Activities page is displayed correctly.	The activities page loads and is accessible.	Pass	Page loads correctly and is accessible.
008	Verify that the True or False quiz displays the correct questions and available answers.	The user has selected the game from the activities page.	Select the game from the activities page.	The quiz is displayed correctly.	The quiz loads as expected.	Pass	Works fine and is randomised on each loading of the quiz.
009	Verify that the True or False quiz displays the correct output if the answer is correct.	The user has chosen correct answers.	Click the correct answers and click 'Submit'.	The 'Correct' answer display message is presented.	The quiz works as expected.	Pass	Works correctly.
010	Verify that the True or False quiz displays the correct output if the answer is incorrect.	The user has chosen incorrect answers.	Click the incorrect answers and click 'Submit'.	The 'Incorrect' answer display message is presented.	The quiz works as expected.	Pass	Works correctly.
011	Verify that the results page for the True or False quiz is presented.	The user has submitted their answers.	Complete the quiz and see if the results page is loaded.	The results page is loaded.	The results page works as expected.	Pass	Works correctly after the 10 questions answered.
012	Verify that the learning recommendations output is present on the True or False results page.	The user has been given a score for their performance in the quiz	Complete the quiz, and check the output of recommendation	The correct learning recommendation is presented on the True or False quiz page.	N/A	Pass	Has been noted from the first deliverable but was not implemented as part of the final delivered tool.
013	Verify that the Multiple Choice quiz displays the correct questions and available answers.	The user has selected the game from the activities page.	Select the game from the activities page.	The quiz is displayed correctly.	The quiz works as expected.	Pass	Displays as intended.
014	Verify that the Multiple Choice quiz displays the correct output if the answer is correct.	The user has chosen correct answers.	Click the correct answers and click 'Submit'.	The 'Correct' answer display message is presented.	The quiz works as expected.	Pass	Displays as intended.

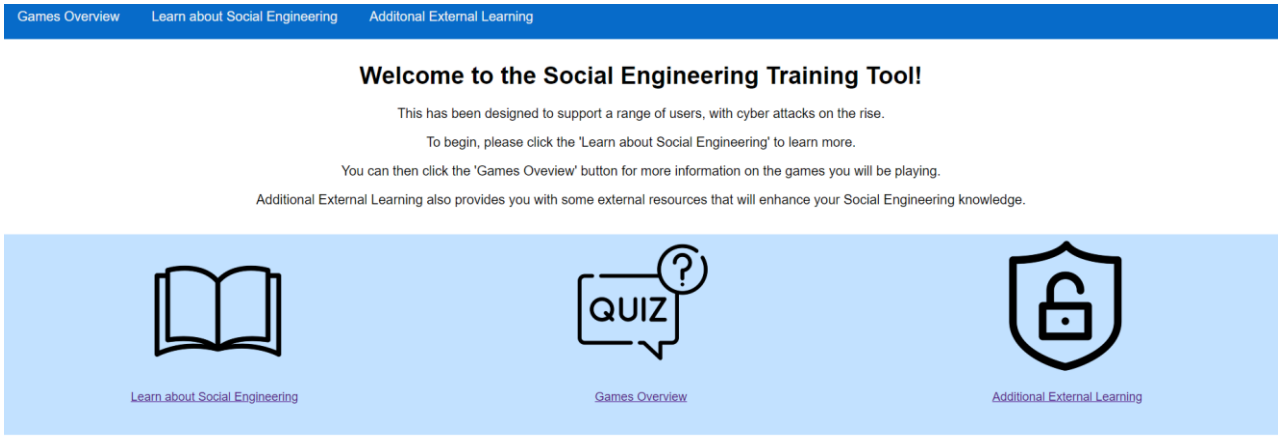
015	Verify that the Multiple Choice quiz displays the correct output if the answer is incorrect.	The user has chosen incorrect answers.	Click the incorrect answers and click ‘Submit’.	The ‘Incorrect’ answer display message is presented.	The quiz works as expected.	Pass	Displays as intended.
016	Verify that the results page for the Multiple Choice quiz is presented.	The user has submitted their answers.	Complete the quiz and see if the results page is loaded.	The results page is loaded.	The quiz works as expected.	Pass	Displays as intended.
017	Verify that the learning recommendations output is present on the Multiple Choice results page.	The user has been given a score for their performance in the quiz.	Complete the quiz and check the output of recommendation.	The correct learning recommendation is presented on the Multiple Choice quiz page.	N/A	N/A	Has been noted from the first deliverable but was not implemented as part of the final delivered tool.
018	Verify that the Spot the Difference quiz displays the correct questions and available answers.	The user has selected the game from the activities page.	Select the game from the activities page.	The quiz is displayed correctly.	N/A	N/A	Has been noted from the first deliverable but was not implemented as part of the final delivered tool.
019	Verify that the Spot the Difference quiz displays the correct output if the answer is correct.	The user has chosen correct answers.	Click the correct answers and click ‘Submit’.	The ‘Correct’ answer display message is presented.	N/A	N/A	Has been noted from the first deliverable but was not implemented as part of the final delivered tool.
020	Verify that the Spot the Difference quiz displays the correct output if the answer is incorrect.	The user has chosen incorrect answers.	Click the incorrect answers and click ‘Submit’.	The ‘Incorrect’ answer display message is presented.	N/A	N/A	Has been noted from the first deliverable but was not implemented as part of the final delivered tool.
021	Verify that the results page for the Spot the	The user has submitted	Complete the quiz and see if	The results page is loaded.	N/A	N/A	Has been noted from the first

	Difference quiz is presented.	their answers.	the results page is loaded.				deliverable but was not implemented as part of the final delivered tool.
022	Verify that the learning recommendations output is presented on the Spot the Difference results page.	The user has been given a score for their performance in the quiz.	Complete the quiz and check the output of recommendation.	The correct learning recommendation is presented on the Spot the Difference quiz page.	N/A	N/A	Has been noted from the first deliverable but was not implemented as part of the final delivered tool.
023	Verify that the Multiple Choice prompt page is displayed correctly	The user has selected a quiz and is ready to start	Test the page displays	The page is displayed correctly.	The page loads correctly.	Pass	Loads with the buttons to navigate.
024	Verify that the True or False prompt page is displayed correctly	The user has selected a quiz and is ready to start	Test the page displays	The page is displayed correctly.	The page loads correctly.	Pass	Loads with the buttons to navigate.

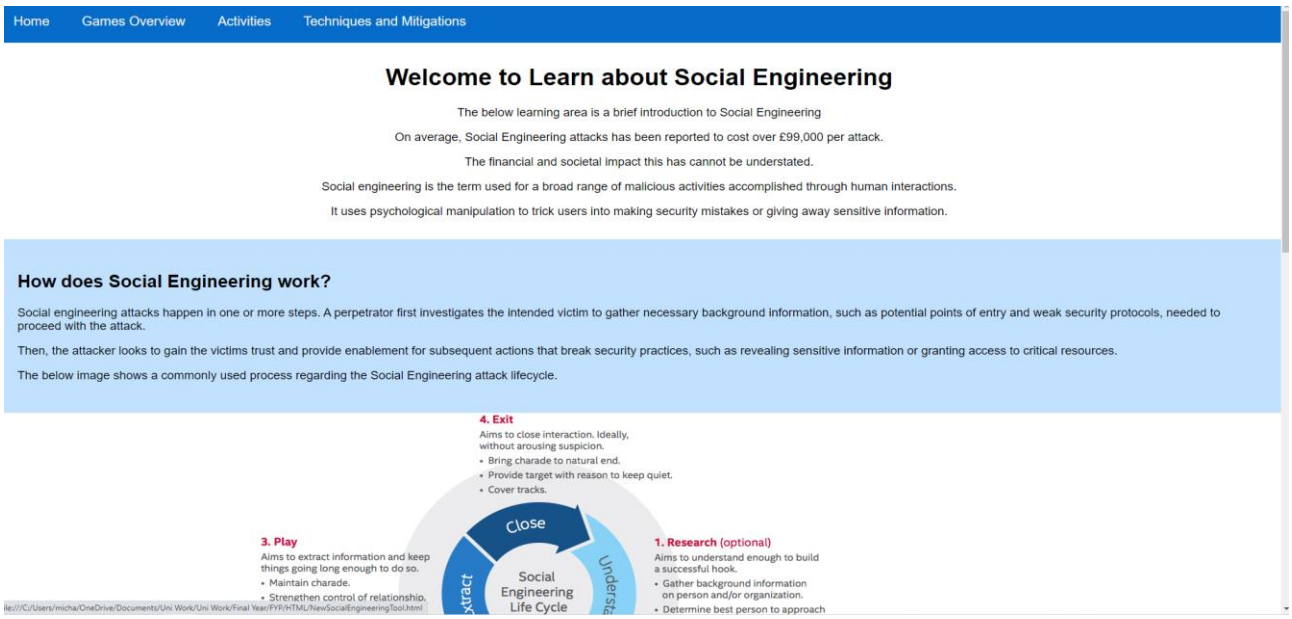
Appendix B

Application:

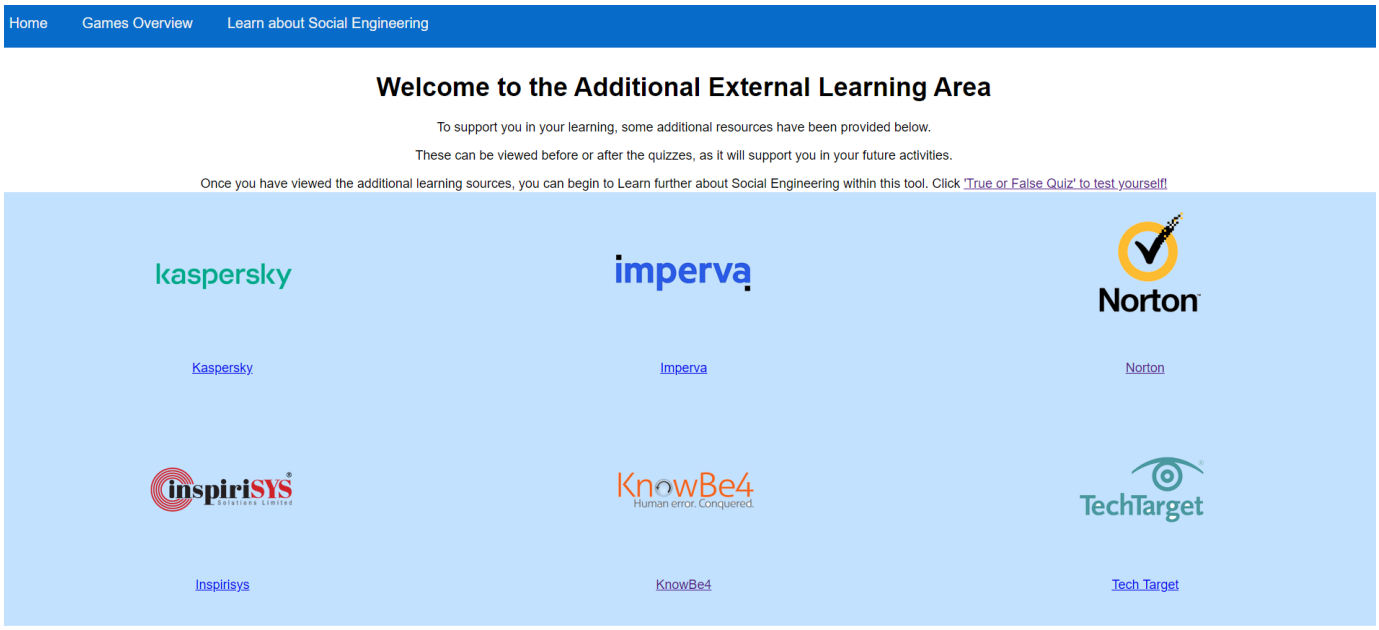
Welcome Page:



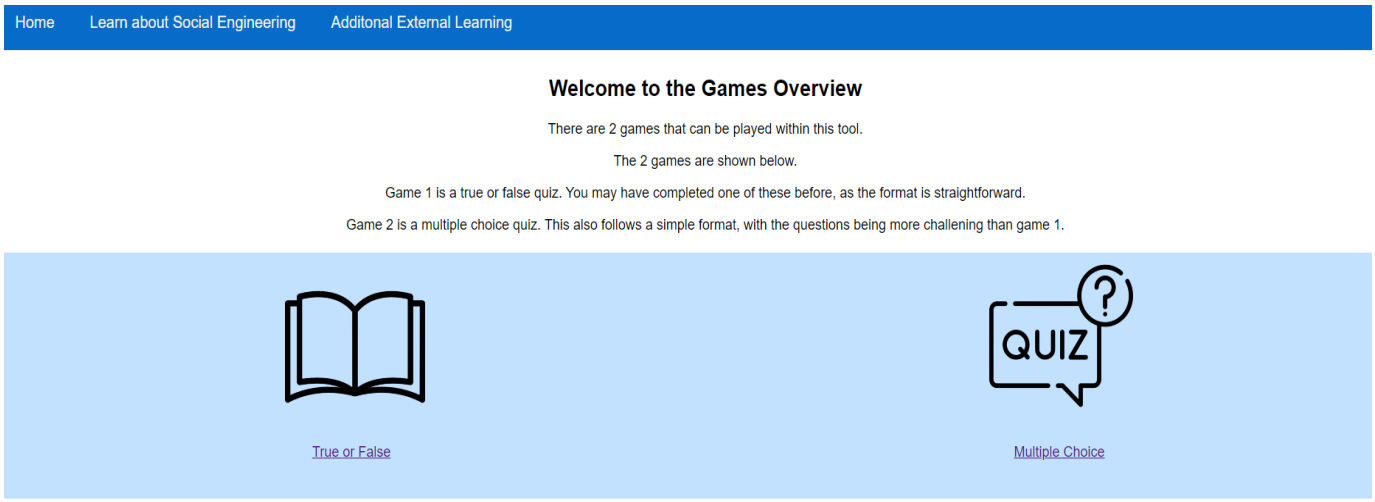
Social Engineering Overview Page:



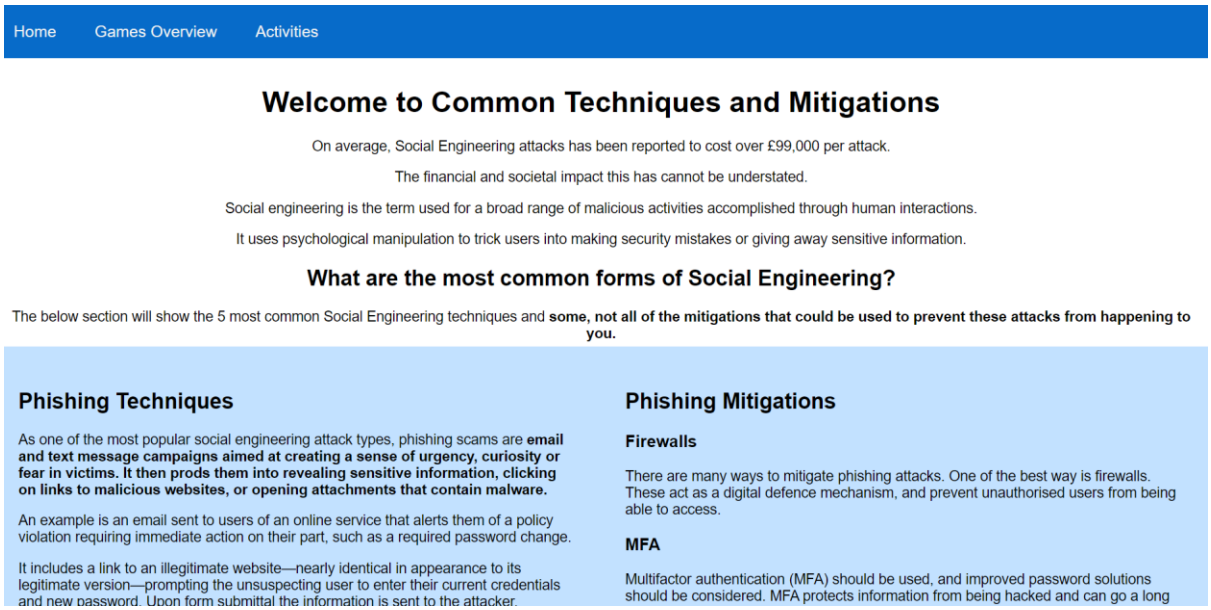
Additional External Learning Page:



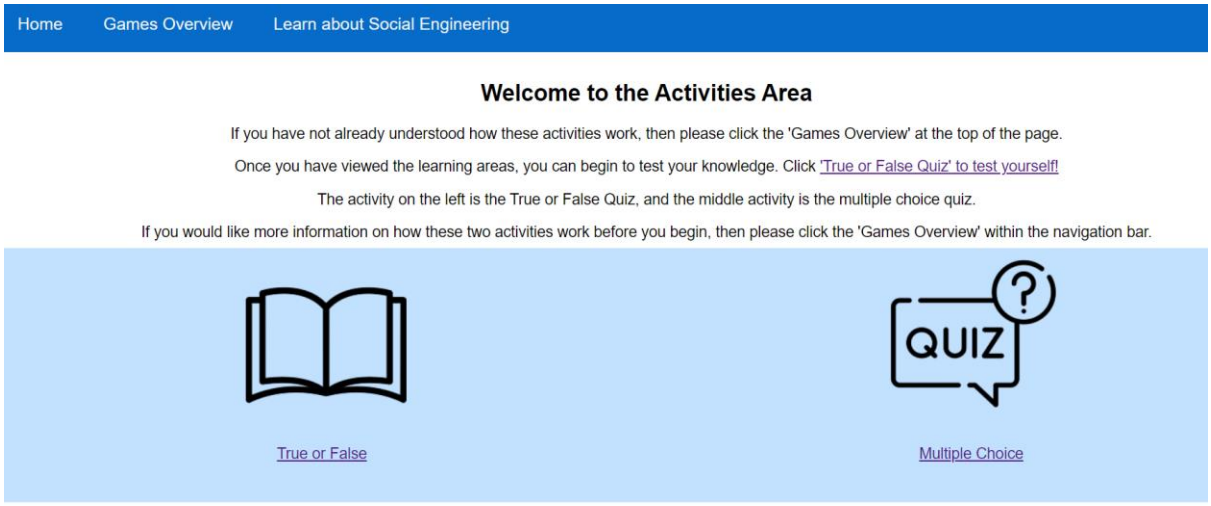
Games Overview Page:



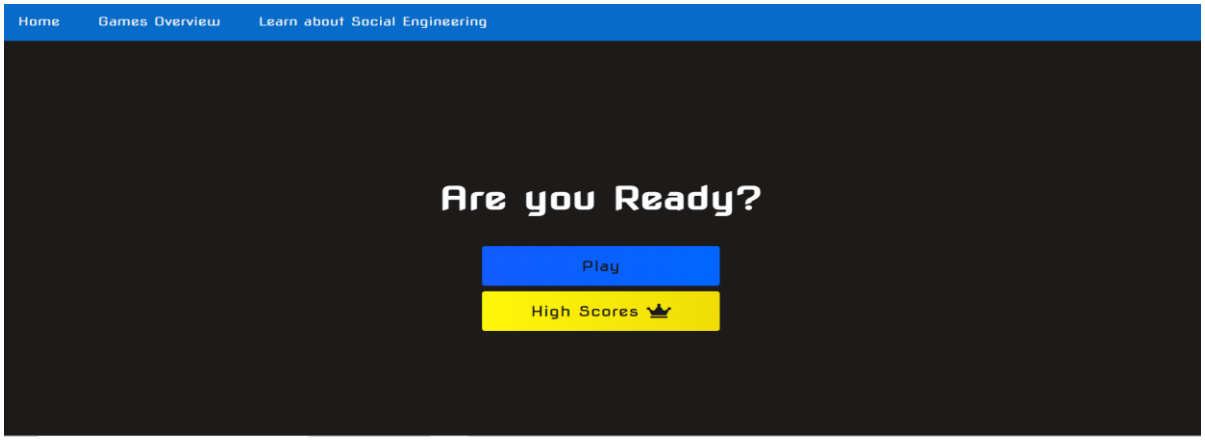
Common Techniques and Mitigations Page:



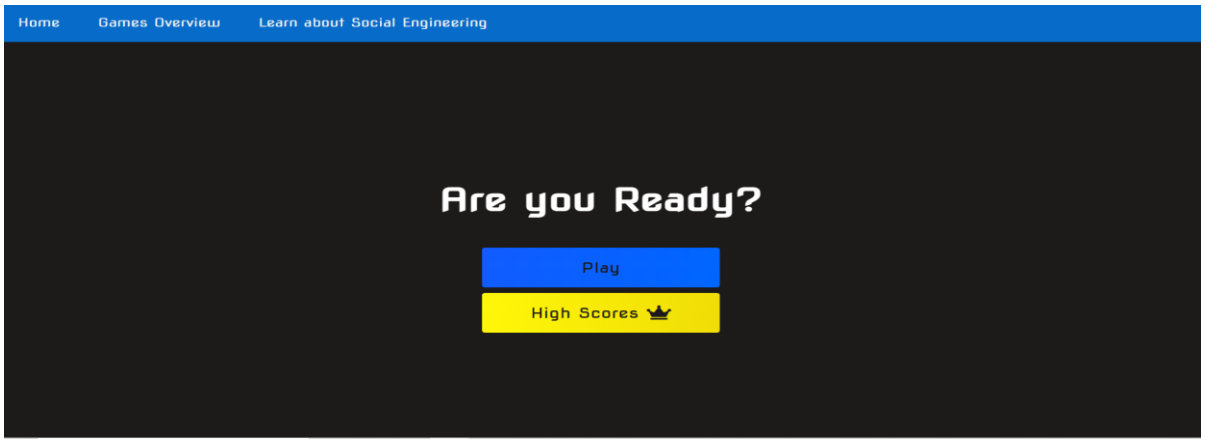
Activities Page:



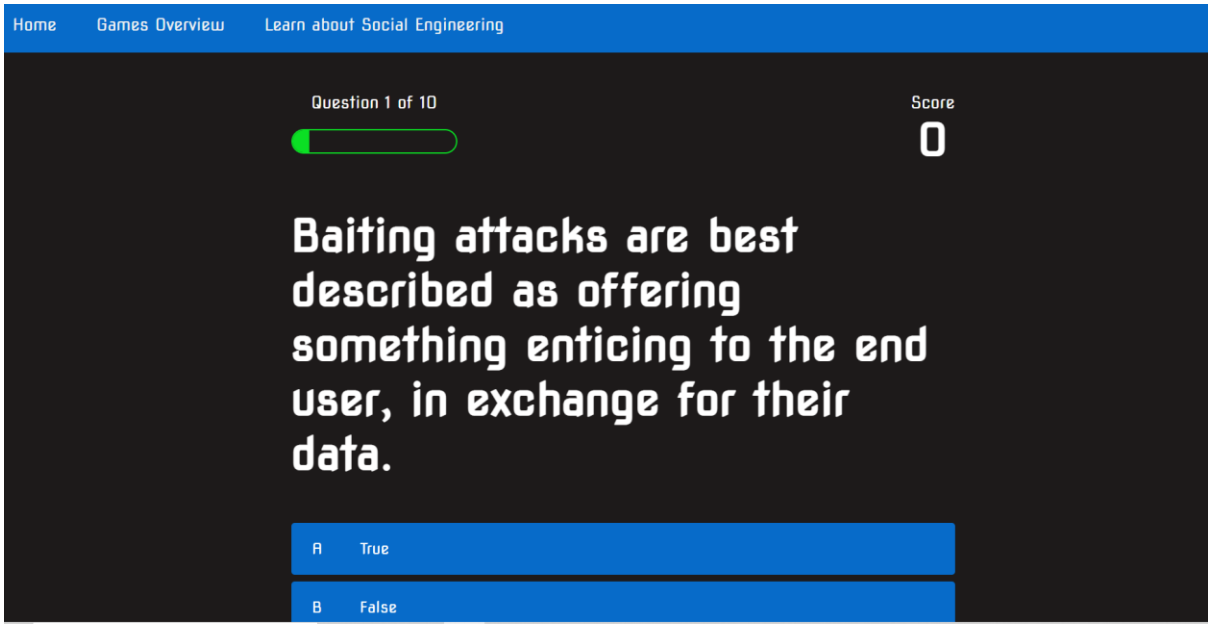
True or False Prompt Page:



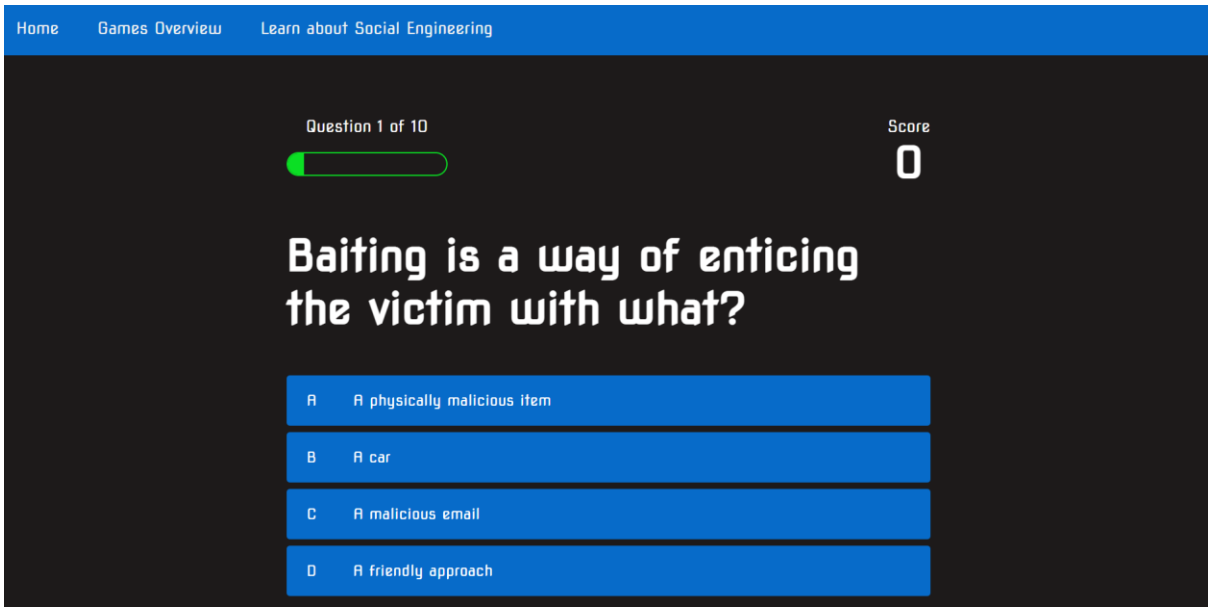
Multiple Choice Prompt Page:



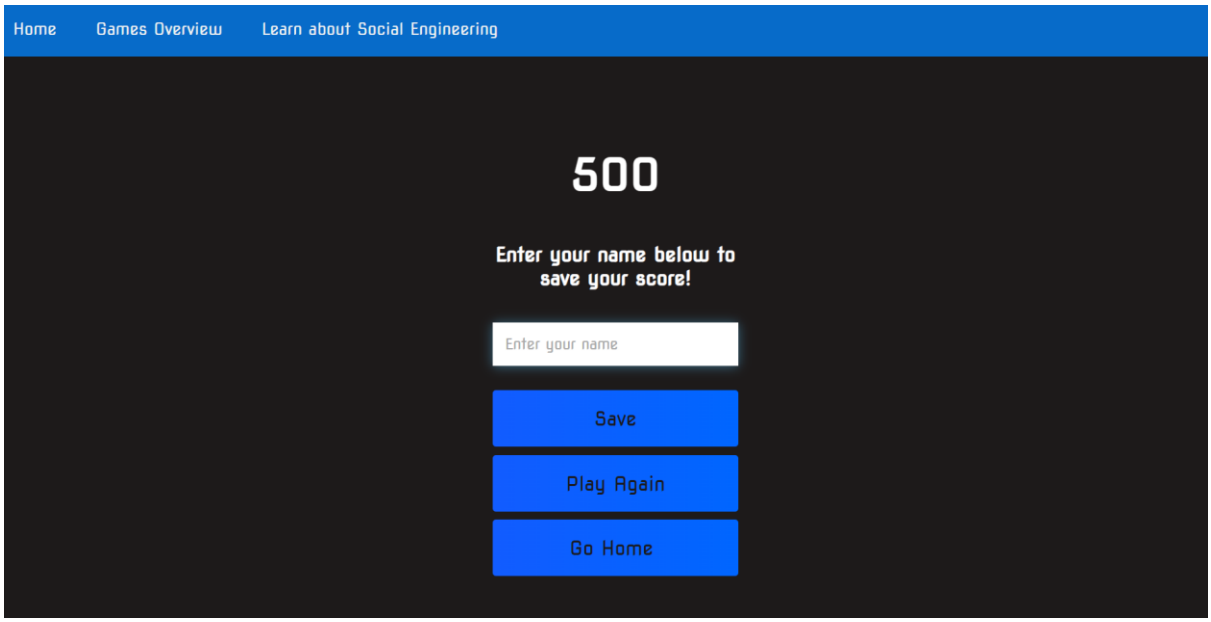
True or False Quiz Page:



Multiple Choice Quiz Page:



True or False Results Page:



Multiple Choice Results Page: