

National College of Ireland

<BSc (Honours) in Computing>

<Cybersecurity>

<2022/2023>

<Seán Mc Donnell>

<x19490822>

<x19490822@student.ncirl.ie>

<"Phish Off" – Don't Get Hooked!>

Final Report Document

Contents

Executive Summary	2
1.0 Introduction	2
1.1. Background	2
1.2. Aims.....	2
1.3. Technology.....	3
1.4. Software Development Methodology	3
1.5. Structure	4
2.0 System.....	4
2.1. Requirements.....	4
2.1.1. Functional Requirements.....	4
2.1.1.1. Use Case Diagram	5
2.1.1.2. Requirement 1: Access Phishing Game 'Email/SMS'	5
2.1.1.3. Description & Priority.....	5
2.1.1.4. Use Case	5
2.1.1.5. Requirement 2: Access Phishing Game 'Quiz'	7
2.1.1.6. Description & Priority.....	7
2.1.1.7. Use Case	7
2.1.1.8. Requirement 3: Access Phishing Game 'Voice Call/Vishing'	9
2.1.1.9. Description & Priority.....	9
2.1.1.10. Use Case	9
2.1.2. Data Requirements	11
2.1.3. User Requirements	11
2.1.4. Environmental Requirements	11
2.1.5. Usability Requirements.....	11
2.2. Design & Architecture	12
2.3. Implementation	12
2.4. Graphical User Interface (GUI).....	15
2.5. Testing.....	20
2.6. Evaluation	23
3.0 Conclusions	23
4.0 Further Development or Research	24
5.0 References	25
6.0 Appendices.....	27
6.1. Project Proposal	27
13.1. Reflective Journals	34

Executive Summary

The purpose of this report is to detail the inspiration, research, goals, planning, technology and strategies of testing for this 4th Year Computing Project. The cyber threat of Phishing is becoming more and more prominent as a danger to the privacy of online users and the integrity of their systems. It is not widely known among the public and therefore many people are not aware that by simply clicking a link from an email, they have released harmful software onto their system. This project aims to develop an interactive application meant to increase user awareness of Phishing through its functionality as well as providing a text list of safety precautions and recommendations of other sources helpful for preventing or avoiding such attacks.

I have researched similar existing tools that provide training for recognising Phishing threats and I believe my application is different and novel in the way the training is set out for the user in the 3 Phishing Games. For example, it is very easy to navigate and not over-saturated with content making it difficult to remember where to access certain functionalities. An example of a similar application for Phishing education is EdApp, which allows you to create a custom phishing training course and provides a comprehensive course library that covers many cyber threats in addition to Phishing (edapp.com, 2022). I see my application being innovative in that it provides detailed documentation on Phishing and links to external research as well as the interactive Phishing Games, making it a multipurpose application.

We will discuss what inspired the idea for this application, the problems it intends to solve, and the various tools and resources used to bring it to fruition. We will then provide a Use Case Diagram, complete with descriptions and priorities, as well as discuss the different requirements that it should meet, i.e. Data and Environmental. Code snippets and screenshots of the running application will also be provided and described in detail. Testing strategies, tools and results will also be covered.

Lastly, we will give a summary of the strengths and limitations as well as possible actions of how to expand the application in terms of research and functionality.

1.0 Introduction

1.1. Background

As people continuously engage with digital devices in their everyday lives, the possibility of having data stolen or compromised is an increasingly prominent threat. During my 3rd year work experience placement in a health facility, there was a proactive approach taken to alert employees to the cybersecurity threat of Phishing. I became aware that the general public may not have enough awareness of the threat of Phishing. Therefore, I chose this as the topic for my 4th year Computer Project.

1.2. Aims

This Project aims to increase cybersecurity awareness through an interactive application meant to inform, educate, and help users to identify the different types of Phishing attacks

and the countermeasures to be taken in the event of such an attack. This application is user friendly and suitable for all ages. Many people do not fully recognise what Phishing is and this app helps users to recognise the traits of a Phishing attack in an email or otherwise. Phishing refers to the act of a malicious user impersonating a trustworthy source in order to steal personal information from an individual or company by deploying malicious software. This type of attack is delivered mostly in the form of emails, as well as advertisements in trustworthy websites (cofense.com, 2022).

1.3. Technology

My implementation language of choice for this project is Python, which is popular for programming web applications due to an easy to understand syntax (devops.com, 2022). It is also not restricted depending on what kind of application is being designed and can be used as per the specific requirements, according to (www.upgrad.com, 2020). I installed and utilised the Python based framework Django to assist in the procedure by creating a virtual environment to run the application in my chosen Browser. Along with the Python code, it features the use of HTML to develop the skeleton for the different pages of the application, as well as provide appropriate links to those pages and the external recommended reading. CSS is also utilised to customise the text colour and font of the different pages. It also includes the use of Bootstrap for responsive navigation and web pages. I used Visual Studio (VS) Code as my IDE to design the application and made sure to push the code to my GitHub repository to supply backup code in the event of accidental deletion or misplacement of the code.

1.4. Software Development Methodology

When planning and developing the application, I followed the Waterfall methodology and the associated stages including:

- Requirement Gathering: Here, I sourced various articles to acquire and document information about what was required for the application, i.e. the common traits of a Phishing attack.
- System Design: I decided on what would be my choice of programming languages, frameworks, testing software and other tools and techniques based on the requirements gathered earlier.
- Implementation: I began the process of coding Python, HTML and CSS into my application in VS Code following the installation of the Django framework and creation of the virtual environment. The implementation of Django was the first step taken in this stage.
- Testing: Once I had achieved a satisfactory level of functionality for my application, I set up test cases to perform 'pytest' in the Git Bash terminal as Automated Testing. I also performed Manual Testing by making a list of tasks to execute in the application and determining if they were as pass or a fail. The results of both types of Testing were satisfactory.
- Delivery: Upon completion of the project, I submitted a zipped folder for the code along with the report document, PowerPoint presentation, poster and links to the video presentation and GitHub code repository in a text file to Moodle.

- **Maintenance:** Throughout the entire coding process, I regularly pushed the code to my GitHub repository to provide security in case of accidental data destruction. Additionally, in the event where I am advised there may be an issue with my application to address, I will search for the solution online (lucidchart.com, 2023).

1.5. Structure

In Section 1 we discuss what the project aims to do to fulfil its role as a Cybersecurity application as well as the technology and coding languages utilised to implement it. We also discuss the rising problems that served as the inspiration for the development of this project.

In Section 2 we discuss the various requirements the project must be able to fulfil, which includes what tools and frameworks are needed, the Use Case Diagram and detailed descriptions of the different Use Cases, how easy the application should be to use and understand as well as what are the necessary conditions to allow for its operation. We also give detail of how the application will look with code snippets, screenshots of the working application, diagrams and strategies for testing.

2.0 System

2.1. Requirements

All requirements should be verifiable. For example, experienced controllers shall be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by experienced users shall not exceed two per day.

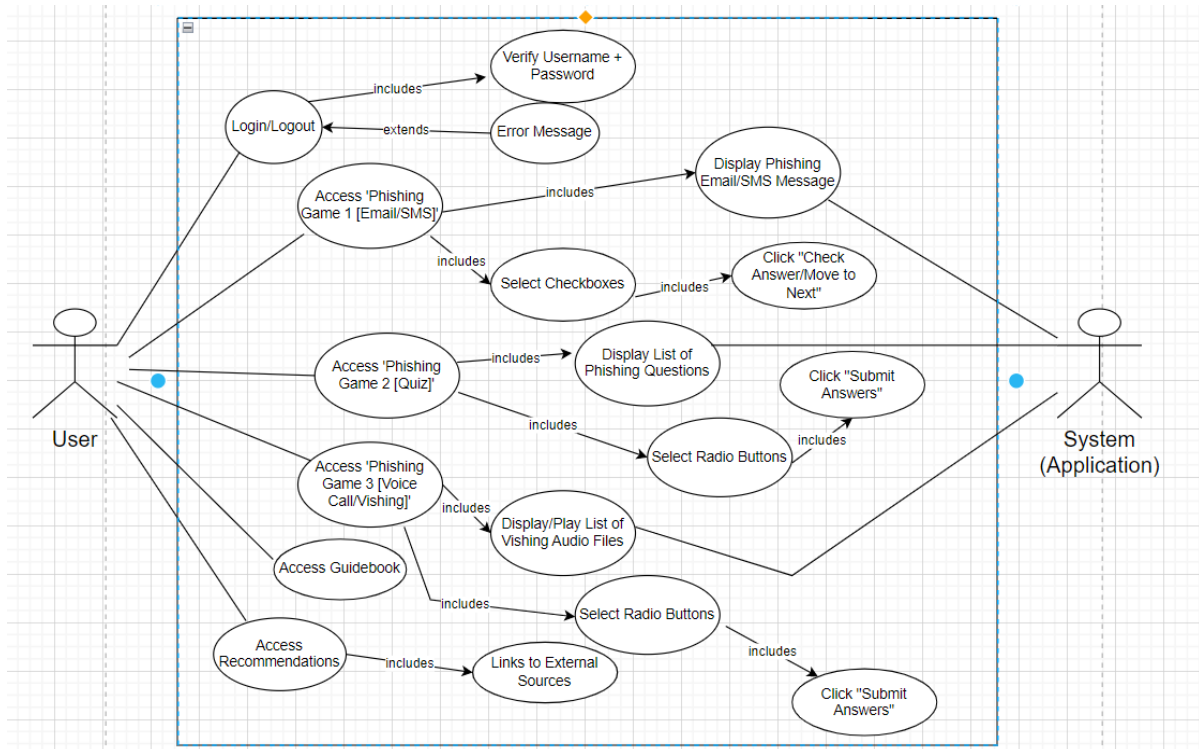
2.1.1. Functional Requirements

This section lists the functional requirements in **ranked order**. Functional requirements describe the possible effects of a software system. In other words, they describe what the system must accomplish. Other kinds of requirements (such as interface requirements, performance requirements, or reliability requirements) describe how the system accomplishes its functional requirements. Each functional requirement is specified as follows:

- ➔ **High Priority:** Phishing Game 1 must show images of Phishing emails and text messages as well as 4 checkboxes below each. Only when at least one of the correct answers and none of the incorrect answers have been selected when the user hits submit can they move to the next question in the session.
- ➔ **High Priority:** Phishing Game 2 must display a list of multiple choice questions and the user must pick one out of four answers for each. Upon hitting the submit button, the application displays a message showing how many questions were correctly answered.
- ➔ **High Priority:** Phishing Game 3 must play several sample voice/phone calls and the user must determine if it is a Phishing call or not.
- ➔ **High Priority:** The application must allow the user to select a Phishing Game from the Homepage.

- ➔ **Medium Priority:** The application must allow the user to access the Homepage, Guidebook, and Recommendations via the Navbar from the Menu.
- ➔ **Medium Priority:** The application must open the external sources from the Recommendations page in a new tab when they are selected.
- ➔ **Low Priority:** The application must allow the user to log in with a username and password in the Login/Logout page.

2.1.1.1. Use Case Diagram



2.1.1.2. Requirement 1: Access Phishing Game 'Email/SMS'

2.1.1.3. Description & Priority

- ➔ A link on the Homepage is provided that will direct the user to the first of three Phishing Games.
- ➔ When in use an image of a Phishing message, either in the form of an email or text message, will be displayed and the user will be asked to select the buttons that correspond to the traits of Phishing.
- ➔ There are several questions to be answered in a session.
- ➔ The navbar is absent here so that the user is less likely to accidentally exit during a session.
- ➔ This requirement occupies a key element of the applications' established functionality, to provide education on the threat of Phishing through an interactive experience, and must be given the same level of high priority as the other 2 Phishing Games.

2.1.1.4. Use Case

Use Case ID: UC-1 v1.1

Scope

The scope of this use case is to educate the user on how to thoroughly scan emails and text messages themselves to identify elements of a possible Phishing attack, e.g. spelling and grammar errors.

Description

This use case describes how to access the first Phishing Game from the Homepage, how to properly use it during a session and what to be aware of when interacting with it.

Flow Description

Precondition

The user must currently be on the Homepage when the application is running.

Activation

This use case starts when the user clicks the link on the Homepage titled "Phishing Game 1 [Email/SMS]".

Main flow

1. A1 <User> clicks the uppermost link on the Homepage.
2. E1 <Application> directs A1 <User> to the interactive Phishing Game.
3. E1 <Application> displays an example Phishing attack in the form of either a text message or email. There are 4 example messages per session.
4. A1 <User> selects the checkboxes below that describe elements of a Phishing attack present within the message.
5. A1 <User> selects 'Check Answer/Move to Next!' and continues through the session.
6. E1 <Application> displays the answer to the previous question at the top of the page.
7. A1 <User> selects 'Check Answer/Finish!' upon submitting their answer for the last example message.
8. A1 <User> is returned to the Homepage.

Alternate flow

A1 : <User>

1. A1 <User> clicks the uppermost link on the Homepage.
2. E1 <Application> directs the user to the interactive Phishing Game.
3. E1 <Application> displays an example Phishing attack in the form of either a text message or email. There are 4 example messages per session.
4. A1 <User> selects any checkboxes that correspond to something that is not a sign of a Phishing attack, e.g. correct spelling or proper acknowledgement of the company the sender works for.
5. A1 <User> selects submit.

6. E1 <Application> displays text providing the user with hints for what to be looking for in the above message.
7. A1 <User> remains on the same question until one correct answer and no incorrect answers are selected upon hitting submit.

Exceptional flow

E1 : <Application>

1. A1 <User> clicks the uppermost link on the Homepage.
2. E1 <Application> directs A1 <User> to the interactive Phishing Game.
3. E1 <Application> displays an example Phishing attack in the form of either a text message or email. There are 4 example messages per session.
4. A1 <User> clicks submit immediately without clicking any of the buttons on the example message.
5. A1 <User> remains on the same page.

Termination

The user selects the prompt labelled 'Check Answer/Finish!' upon submitting a correct answer for the last example message.

Post condition

The user is returned to the Homepage.

[2.1.1.5. Requirement 2: Access Phishing Game 'Quiz'](#)

[2.1.1.6. Description & Priority](#)

- ➔ A link on the Homepage is provided that will direct the user to the second of three Phishing Games.
- ➔ When in use, the user will see a list of questions on their screen that relate to Phishing.
- ➔ They will have a choice of 4 different answers for each and must select one.
- ➔ There are several questions to be answered in a session.
- ➔ The navbar is absent here so that the user is less likely to accidentally exit during a session.
- ➔ This requirement occupies a key element of the applications' established functionality, to provide education on the threat of Phishing through an interactive experience, and must be given the same level of high priority as the other 2 Phishing Games.

[2.1.1.7. Use Case](#)

Use Case ID: UC-2 v1.1

Scope

The scope of this use case is to educate users on how to identify the various types of Phishing attacks, as well as gain a clear understanding of how to avoid and counteract them.

Description

This use case describes how to access the second Phishing Game from the Homepage, how to properly use it during a session and what to be aware of when interacting with it.

Flow Description

Precondition

The user must currently be on the Homepage when the application is running.

Activation

This use case starts when the user clicks the link on the Homepage titled “Phishing Game 2 [Quiz]”.

Main flow

1. A1 <User> clicks the middle link on the Homepage.
2. E1 <Application> directs A1 <User> to the interactive Phishing Game.
3. E1 <Application> presents A1 <User> with a list of questions relating to Phishing attacks and presents 4 possible answers to select for each. There are a total of 9 questions per session.
4. A1 <User> selects their answer for each by toggling the radio button next to their chosen answer.
5. A1 <User> selects ‘Submit Answers’.
6. E1 <Application> displays the total number of the user’s correct answers.
7. A1 <User> answers all 9 questions correctly.
8. A1 <User> selects ‘Exit’ and is returned to the Homepage.

Alternate flow

A1 : <User>

1. A1 <User> clicks the middle link on the Homepage.
2. E1 <Application> directs A1 <User> to the interactive Phishing Game.
3. E1 <Application> presents A1 <User> with a list of questions relating to Phishing attacks and presents 4 possible answers to select for each. There are a total of 9 questions per session.
4. A1 <User> selects their answer by toggling the radio button next to their chosen answer.
5. A1 <User> selects ‘Submit Answers’.
6. E1 <Application> displays the total number of the user’s correct answers.
7. A1 <User> selects ‘Exit’ and is returned to the Homepage.

Exceptional flow

E1 : <Application>

1. A1 <User> clicks the middle link on the Homepage.
2. E1 <Application> directs A1 <User> to the interactive Phishing Game.
3. E1 <Application> presents A1 <User> with a list of questions relating to Phishing attacks and presents 4 possible answers to select for each. There are a total of 10 questions per session.

4. A1 <User> clicks 'Submit Answers' without having selected any of the radio buttons.
5. E1 <Application> displays a message letting them know that they must not leave anything blank.

Termination

The user selects the prompt labelled 'Submit Answers' upon submitting an answer for all 9 questions.

Post condition

The user returns to the Homepage.

[2.1.1.8. Requirement 3: Access Phishing Game 'Voice Call/Vishing'](#)

[2.1.1.9. Description & Priority](#)

- ➔ A link on the Homepage is provided that will direct the user to the third of three Phishing Games.
- ➔ When in use, the user will see several audio files with text above saying "Play and listen to the Audio Files below and identify which ones are Vishing Calls".
- ➔ They will have to toggle the box for either Yes or No for each and then click Submit.
- ➔ The navbar is absent here so that the user is less likely to accidentally exit during a session.
- ➔ This requirement occupies a key element of the applications' established functionality, to provide education on the threat of Phishing through an interactive experience, and must be given the same level of high priority as the other 2 Phishing Games.

[2.1.1.10. Use Case](#)

Use Case ID: UC-3 v1.1

Scope

The scope of this use case is to educate users on how to recognise when a caller that claims to be a representative of a major company, appears to be alerting the user about a virus or security breach in order to retrieve important information from them.

Description

This use case describes how to access the third Phishing Game from the Homepage, how to properly use it during a session and what to be aware of when interacting with it.

Flow Description

Precondition

The user must currently be on the Homepage when the application is running.

Activation

This use case starts when the user clicks the link on the Homepage titled “Phishing Game 3 [Voice Call/Vishing]”.

Main flow

1. A1 <User> clicks the lowermost link on the Homepage.
2. E1 <Application> directs A1 <User> to the interactive Phishing Game.
3. E1 <Application> presents A1 <User> with a series of audio files and after listening to one, they must determine whether it is or is not a Vishing scam. There are a total of 6 voice samples per session.
4. A1 <User> selects their answer by toggling the radio button next to their chosen answer.
5. A1 <User> selects ‘Submit’.
6. E1 <Application> displays the number of correct answers.
7. A1 <User> answers all questions correctly.
8. A1 <User> selects ‘Exit’ and is returned to the Homepage.

Alternate flow

A1 : <User>

1. A1 <User> clicks the lowermost link on the Homepage.
2. E1 <Application> directs A1 <User> to the interactive Phishing Game.
3. E1 <Application> presents A1 <User> with a series of audio files and after listening to one, they must determine whether it is or is not a Vishing Scam. There are a total of 6 voice samples per session.
4. A1 <User> selects their answer by toggling the radio button next to their chosen answer.
5. A1 <User> selects ‘Submit’.
6. E1 <Application> displays the number of correct answers verses the total number of questions in the session.
7. A1 <User> selects ‘Exit’ and is returned to the Homepage.

Exceptional flow

E1 : <Application>

6. A1 <User> clicks the lowermost link on the Homepage.
7. E1 <Application> directs A1 <User> to the interactive Phishing Game.
8. E1 <Application> presents A1 <User> with a series of audio files and after listening to one, they must determine whether or not it is a Vishing scam. There are a total of 6 voice samples per session.
9. A1 <User> immediately clicks ‘Submit’ without toggling an of the radio buttons or playing any of the files.
10. E1 <Application> displays a message letting them know that they must not leave anything blank and they remain on the same page.

Termination

The user selects the prompt labelled ‘Submit’ upon submitting an answer for all 6 voice samples.

Post condition

The user returns to the Homepage.

2.1.2. Data Requirements

- ➔ Python includes powerful tools used for connecting to and/or creating databases. It supports several quality databases such as SQLite and MySQL (tutorialspoint.com, 2022).
- ➔ I sourced various online articles and websites to collect the information required to compose the questions for each of the 3 Phishing Games. Such information includes the fact that Vishing attack strategies involve impersonating an organisation representative or informing a person that they have won a sum of money (bitlyft.com, 2023). Spear Phishing is a popular attack where offenders claim to be a trusted entity to trick victims into disclosing personal information or download malware (armorblox.com, 2023).
- ➔ The background image for the Homepage and all images used in Phishing Game 1 were sourced from searching images in the Google Chrome browser.
- ➔ The audio files used in Phishing Game 3 were recorded using the 'Voice Recorder' app installed on my computer after a script for each file was written and practised. The information used to write out said script was gathered from online articles.

2.1.3. User Requirements

- ➔ Phishing Game 1 'Email/SMS' requires that the user answers at least one correct answer for each example message to continue through the session.
- ➔ Phishing Game 2 'Quiz' and Phishing Game 3 'Voice Call/Vishing' must evaluate the total number of answered questions once the user has clicked 'Submit Answers' and display a message showing how many were correctly answered out of the total number of questions.

2.1.4. Environmental Requirements

- ➔ As this is a web application, it must be present in a location with internet access to be able to operate.
- ➔ Since a virtual environment has been created for this application, it must be able to run the server with the Django framework.
- ➔ This application has no effect on the external environment.

2.1.5. Usability Requirements

- ➔ The application is easy to use and understand for all users.
- ➔ Each of the Phishing Games are easy work through and learn from with few user errors and all buttons are clearly laid out.
- ➔ The interface is simple and easy to navigate with links being clearly displayed.
- ➔ The login process is as simple as any other.

2.2. Design & Architecture

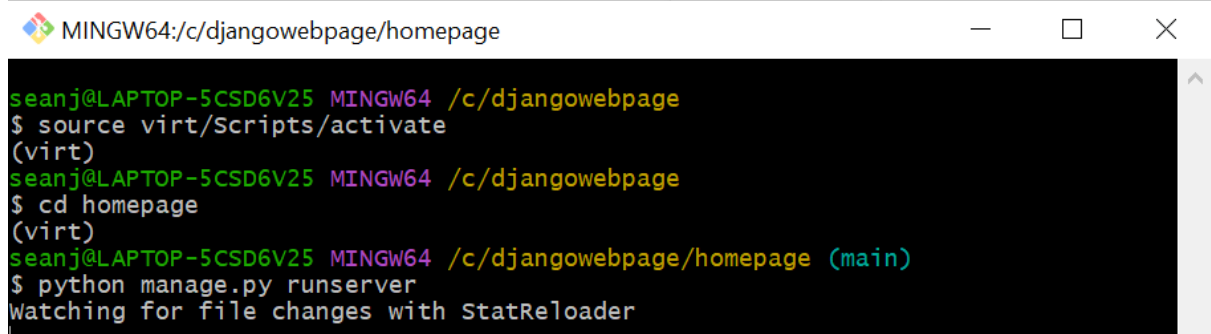
As stated earlier, the Python framework Django is used to create a virtual environment with which to run the application in the Browser. Visual Studio Code is the Integrated Development Environment (IDE) used to develop the Python, HTML and CSS code. Incorporating Django resulted in the generation of several Python files, such as 'admin.py' and 'models.py' which are used for registering and creating models, 'tests.py' and 'urls.py', the latter of which is required to allow the HTML pages to open when the application is running, as well as 'views.py', which allows Python code to provide functionality for those HTML pages. HTML files are included in the 'templates' folder to provide functionality and navigation when using the application, and a 'style.css' file is used to customise font colour and size. This CSS file is included in a static folder where images and audio files must also be placed for them to appear in the Phishing Game pages. Understanding the Static urls and Images in Django and getting them to work properly was a challenge for me, specifically navigating the framework and understanding how the framework manages relative file paths. Additionally, there are several models included in 'models.py' with the intended purpose of creating the database needed to save the user's answers and score when interacting with the Phishing Games. As I was unable to get the models to work, they remain inactive.

2.3. Implementation

When considering which coding languages I should use, I decided to use Python instead of Java or JavaScript as I had worked with Python on a 3rd year project and I felt more competent using it than Java and JavaScript. I was more confident in undertaking the project I envisaged using Python. As I mentioned earlier, Python is a popular choice for web application programming with its user-friendly syntax (devops.com, 2022).

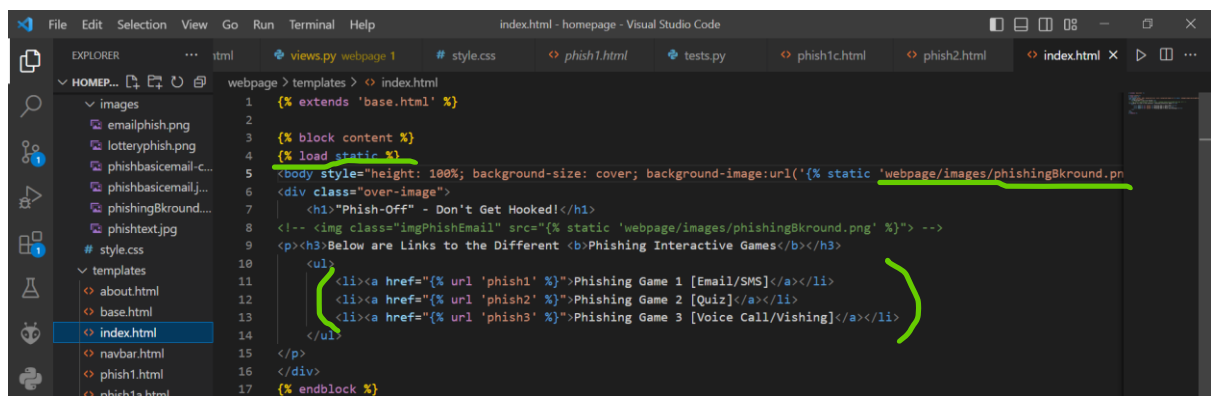
I also utilised a web application framework compatible with the Python language and after having researched my different options, I decided on Django. This framework is based on the Model-View-Template (MVT) software design pattern for web application development (educative.io, 2022). By incorporating this framework, it helped in designing an application that is versatile, being workable in many different formats including HTML, secure, by directly storing passwords, and maintainable, by using design principles and patterns which encourage the creation of reusable code (developer.mozilla.org, 2023). Additionally, I was interested in working with new technology for this project.

Below are the commands in the Git Bash terminal, to activate the virtual environment named 'virt' and run the server in the Browser.



```
MINGW64:/c/djangowebpage/homepage
seanj@LAPTOP-5CSD6V25 MINGW64 /c/djangowebpage
$ source virt/Scripts/activate
(virt)
seanj@LAPTOP-5CSD6V25 MINGW64 /c/djangowebpage
$ cd homepage
(virt)
seanj@LAPTOP-5CSD6V25 MINGW64 /c/djangowebpage/homepage (main)
$ python manage.py runserver
Watching for file changes with StatReloader
```

The following code shows the design of the Homepage, 'index.html'. It is done mainly using the HTML coding language and CSS code is also used to customise the font size and colour in the static file 'style.css'. A URL link to each of the 3 Phishing Games is present in the code along with a background image for the Homepage implemented with the Django command {% load static %}. The Guidebook and Recommendations pages are designed with the same code.



```
index.html - homepage - Visual Studio Code
EXPLORER
  webpage > templates > index.html
  1 {% extends 'base.html' %}
  2
  3 {% block content %}
  4 {% load static %}
  5 <body style="height: 100%; background-size: cover; background-image:url('{% static 'webpage/images/phishingBkround.png' %}')>
  6 <div class="over-image">
  7   <h1>"Phish-Off" - Don't Get Hooked!</h1>
  8 <!-- 
  9 <p><h3>Below are Links to the Different <b>Phishing Interactive Games</b></h3>
 10 <ul>
 11 <li><a href="{% url 'phish1' %}">Phishing Game 1 [Email/SMS]</a></li>
 12 <li><a href="{% url 'phish2' %}">Phishing Game 2 [Quiz]</a></li>
 13 <li><a href="{% url 'phish3' %}">Phishing Game 3 [Voice Call/Vishing]</a></li>
 14 </ul>
 15 </p>
 16 </div>
 17 {% endblock %}
```

The following code shows the implementation, using Python Django, of the views for all the different pages of the application including the Homepage, Guidebook and the 3 Phishing Games in the 'views.py' file. This is required for them to function when the application is being run on the server.

```

1 from django.shortcuts import render, redirect
2 from .models import Sms1
3 #from .forms import Sms1Form
4
5 def index(request):
6     all_members = Sms1.objects.all
7     return render(request, 'index.html', {})
8
9 def about(request):
10    return render(request, 'about.html', {})
11
12 def recc(request):
13    return render(request, 'recc.html', {})
14
15 def phish1(request):
16    if request.method == "POST":
17        form_detail = request.POST
18        print(form_detail)
19        if 'wrong' in [val for val in form_detail.values()]:
20            if 'sender' and 'date' in form_detail.keys():
21                param = {'mesgs': [{"sender": "Clearly check if you can identify the sender!"}, {"date": "Clearly che
22            elif 'sender' in form_detail.keys():
23                param = {'mesgs': [{"sender": "Clearly check if you can identify the sender!"}]}
24            elif 'date' in form_detail.keys():
25                param = {'mesgs': [{"date": "Clearly check that the date matters!"}]}
26
27        return render(request, 'phish1.html', param)
28
29 elif 'wrong' and 'right' not in [val for val in form_detail.values()]:
30    return render(request, 'phish2.html', {})

```

The following code shows the implementation, using HTML, of the navbar used to move between the Homepage, Guidebook, Recommendations and Login/Logout page upon starting up the application. In 'base.html', there is the piece of code {% include 'navbar.html' %}, which makes the navbar visible in the top of the screen from anywhere outside the 3 Phishing Games.

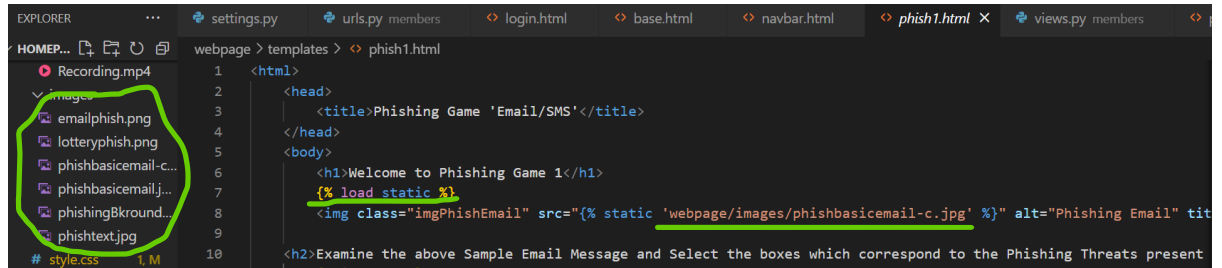
```

1 <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
2   <div class="container-fluid">
3     <a class="navbar-brand" href="{% url 'index' %}">Sean</a>
4     <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"
5       <span class="navbar-toggler-icon"></span>
6     </button>
7     <div class="collapse navbar-collapse" id="navbarSupportedContent">
8       <ul class="navbar-nav me-auto mb-2 mb-lg-0">
9         <li class="nav-item">
10          <a class="nav-link active" aria-current="page" href="{% url 'index' %}">Home</a>
11        </li>
12        <li class="nav-item">
13          <a class="nav-link" href="{% url 'about' %}">Guidebook</a>
14        </li>
15        <li class="nav-item">
16          <a class="nav-link" href="{% url 'recc' %}">Recommendations</a>
17        </li>
18        {% if user.is_authenticated %}
19        <li class="nav-item">
20          <a class="nav-link" href="{% url 'logout' %}">Logout</a>
21        </li>
22        {% else %}
23        <li class="nav-item">
24          <a class="nav-link" href="{% url 'login' %}">Login</a>
25        </li>
26        {% endif %}
27      </ul>
28    <form class="d-flex">
29      <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
30      <button class="btn btn-outline-success" type="submit">Search</button>

```

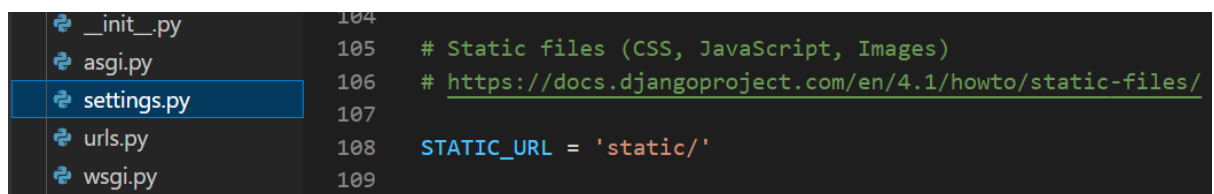
The inclusion of Phishing message images was an important requirement for Phishing Game 1, so the following HTML code and Django command were incorporated into

'phish1.html, phish1a.html, phish1b.html and phish1c.html' to fulfil this requirement.



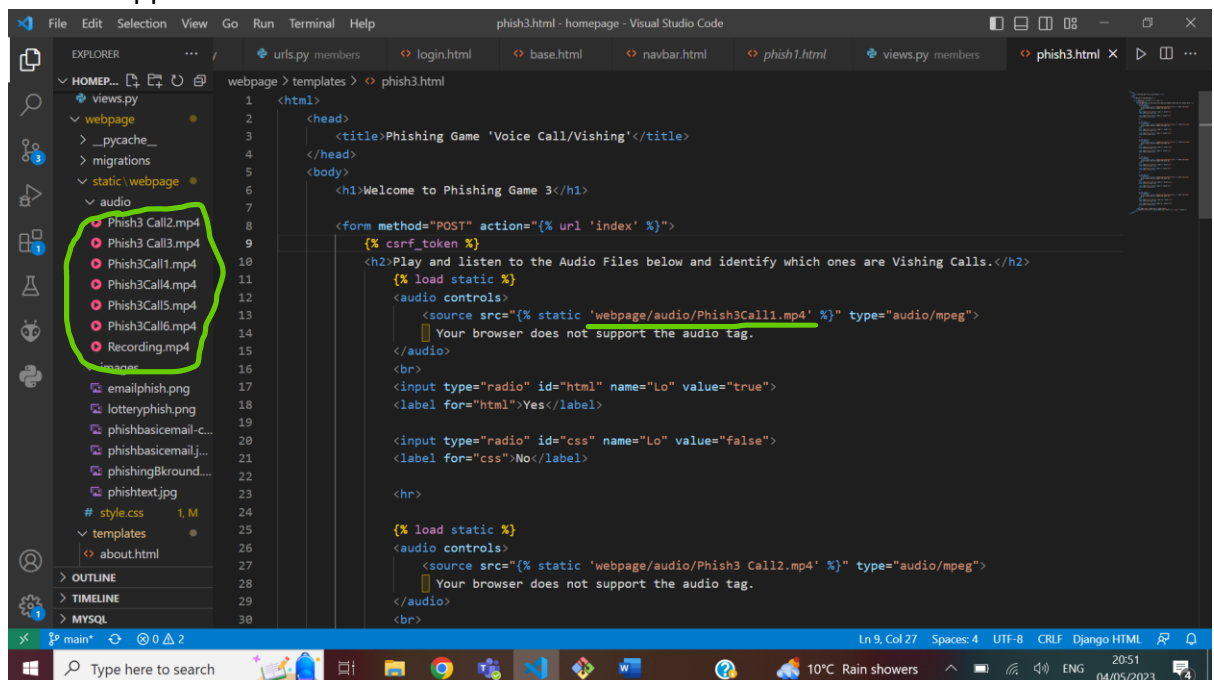
```
1 <html>
2 <head>
3 <title>Phishing Game 'Email/SMS'</title>
4 </head>
5 <body>
6 <h1>Welcome to Phishing Game 1</h1>
7 {% load static %}
8 <img class='imgPhishEmail' src='{% static 'webpage/images/phishbasicemail-c.jpg' %}' alt='Phishing Email' tit
9
10 <h2>Examine the above Sample Email Message and Select the boxes which correspond to the Phishing Threats present
```

Additionally, the images had to be saved within the static folder which also houses the 'style.css' file and their location was to be referenced in the code as shown above. This also applies for the background image mentioned earlier. It was also mandatory to define the static URL in 'settings.py' to allow the images to appear in the running application.



```
104
105 # Static files (CSS, JavaScript, Images)
106 # https://docs.djangoproject.com/en/4.1/howto/static-files/
107
108 STATIC_URL = 'static/'
109
```

Much like the images, the audio files included in Phishing Game 3 were required to be saved into the static folder as well as be defined in the code to work in the running application. The audio was recorded using the 'Voice Recorder' app installed on my computer and had to be converted from M4A to MP4 format before being embedded into the application.

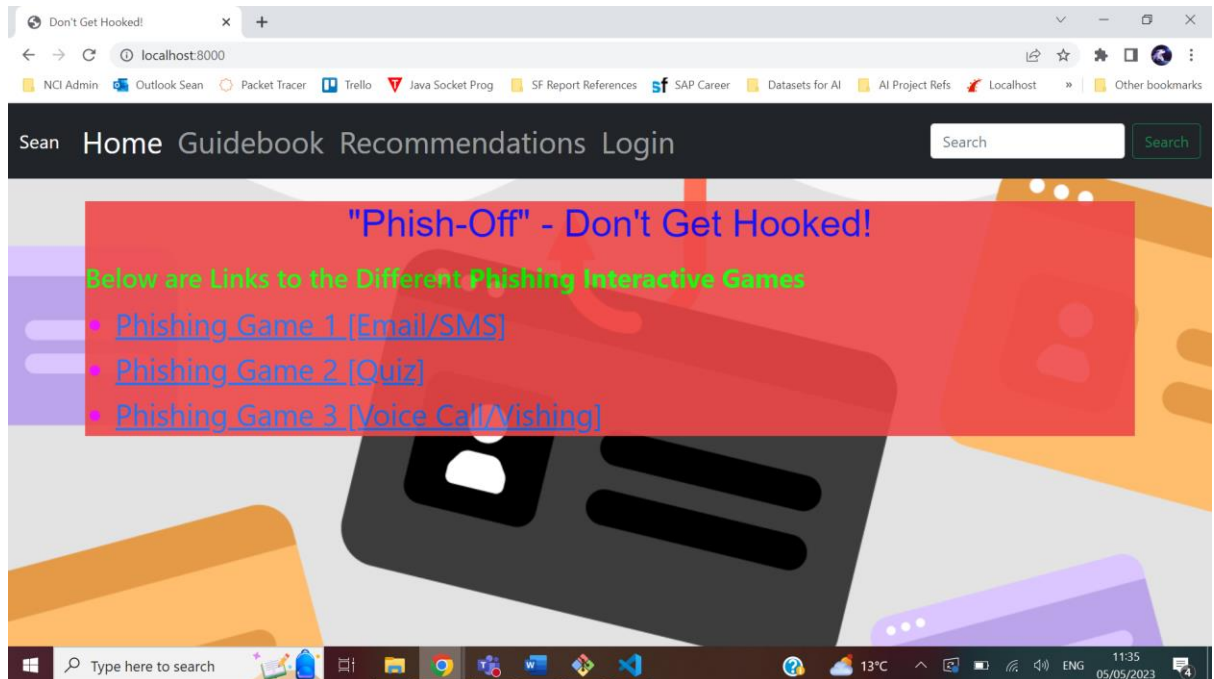


```
1 <html>
2 <head>
3 <title>Phishing Game 'Voice Call/Vishing'</title>
4 </head>
5 <body>
6 <h1>Welcome to Phishing Game 3</h1>
7
8 <form method='POST' action='{% url 'index' %}'>
9 {% csrf_token %}
10 <h2>Play and listen to the Audio Files below and identify which ones are Vishing Calls.</h2>
11 {% load static %}
12 <audio controls>
13 <source src='{% static 'webpage/audio/Phish3 Call1.mp4' %}' type='audio/mpeg'>
14 </audio>
15 Your browser does not support the audio tag.
16 <br>
17 <input type='radio' id='html' name='Lo' value='true'>
18 <label for='html'>Yes</label>
19
20 <input type='radio' id='css' name='Lo' value='false'>
21 <label for='css'>No</label>
22
23 <br>
24
25 {% load static %}
26 <audio controls>
27 <source src='{% static 'webpage/audio/Phish3 Call2.mp4' %}' type='audio/mpeg'>
28 </audio>
29 Your browser does not support the audio tag.
30 <br>
```

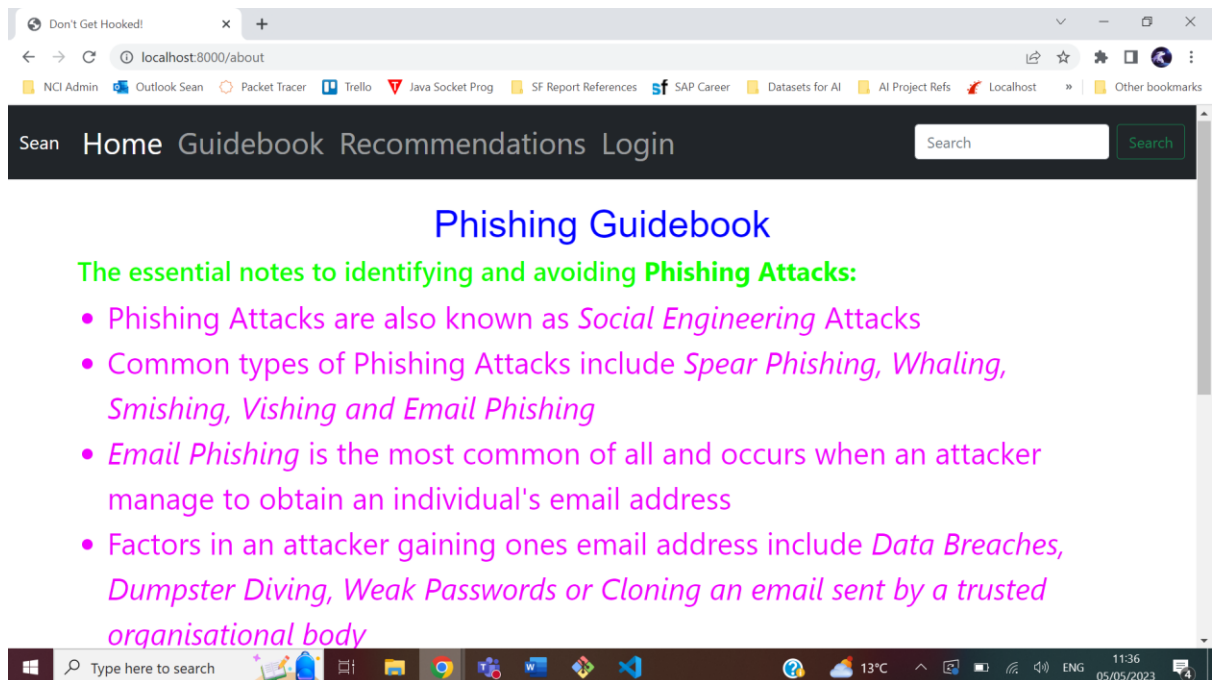
2.4. Graphical User Interface (GUI)

The Homepage is the default page that the user accesses after logging in. This page provides 3 links that direct the user to one of the 3 Phishing Games when clicked on. It features a navbar at the top of the screen which is used to move between the current

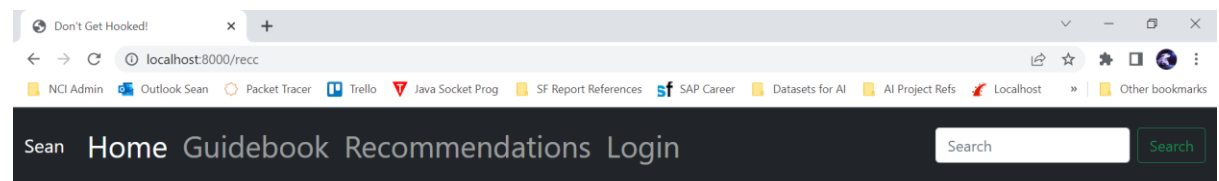
page, the Guidebook, the Recommendations and the Login page. All pages are also responsive including the navbar.



The Guidebook, which is selected via the navbar, displays a detailed list of advice and safety precautions regarding Phishing attacks. It details the different types of Phishing attacks and what they specifically target, various strategies to consider when presented with a Phishing message, and what to do if your system ends up as the victim of a Phishing attack.



The Recommendations, which is selected via the navbar, gives links to multiple articles that describe the do's and don'ts of dealing with Phishing attacks, reports on recent Cyber Attacks which involved Phishing, and Anti Phishing Tools.



Recommended Reading

A List of Links to **Phishing Related Articles** and **Anti Phishing Software**

- [HSE Cyber Attack 2021](#)
- [NHS Ransomware Attack 2022](#)
- [Top 10 Phishing Protection Tools](#)
- [Phishing 101](#)
- [The Danger of Vishing](#)
- [Anti Phishing Techniques](#)



Phishing Game 1, which is accessed through the Homepage, displays an image of a Phishing Email or Text Message and below that features a list of checkboxes with sentences describing the message. Not all of the sentences are true. Only when at least one of the 'right' boxes are selected and the user hits the button labelled 'Check Answer/Move to Next' will it move to the next question.

Phishing Game 'Email/SMS' x +

localhost:8000/phish1

NGI Admin Outlook Sean Packet Tracer Trello Java Socket Prog SF Report References sf SAP Career Datasets for AI AI Project Refs Localhost Other bookmarks

Welcome to Phishing Game 1

Thu 3/14/2019 11:09 AM

KH

New Message From ITS Help Desk

To [redacted]

If there are problems with how this message is displayed, click here to view it in a web browser.
Click here to download pictures. To help protect your privacy, Outlook prevented automatic download of some pictures in this message.

Dear Student

On 13th march 2019, We received a request to close your account, Your request is being processed and all associated data will be removed from our server shortly.

If you made this request, you are all set and no further action is required from you

If not you, please [Click Here](#) to reconfirm your account before 48hrs

Examine the above Sample Email Message and Select the boxes which correspond to the Phishing Threats present in the message.

Tips:

2

ITS Help Desk is the subject and not the sender!

Notice how the message says before 48 hours!

The identity of the sender of this email is clear.

The text in the main body projects a sense of urgency.

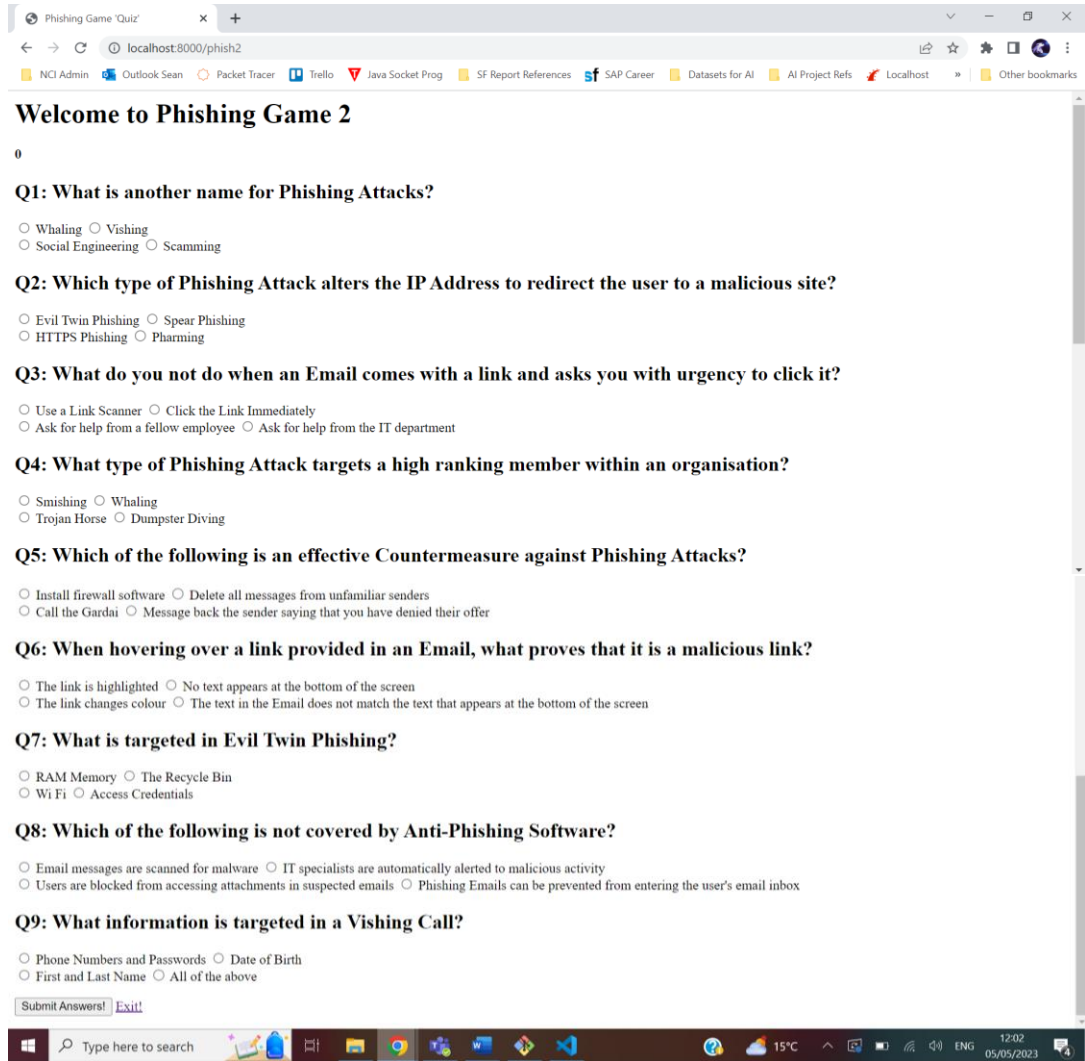
Say the message was just sent to you today. The time in which it was sent is important.

It's okay to click on the link provided.

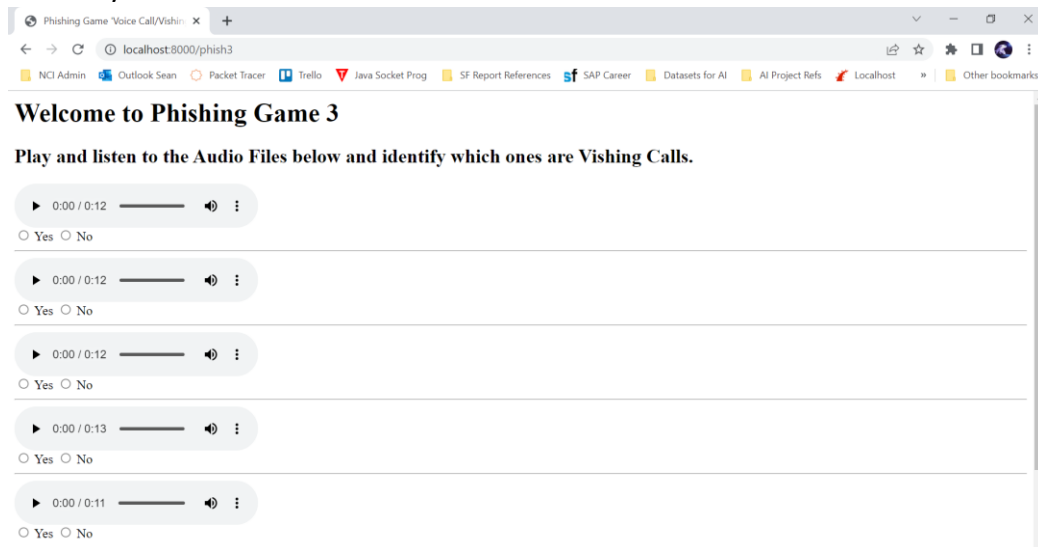
Check Answer Move to Next! Exit!

Type here to search 13°C 11:54 05/05/2023

Phishing Game 2, which is accessed through the Homepage, displays a list of 9 multiple choice questions. When the user hits 'Submit Answers', all the correctly answered questions out of the total number of questions is displayed.



Phishing Game 3, which is accessed through the Homepage, displays several audio files with 'Yes' and 'No' buttons. The user will hear simulations of Vishing calls when they play the audio and when they hit 'Submit', they will see how many they answered correctly.





2.5. Testing

I utilised and combination of Manual and Automated testing for my application. Manual testing is carried out by human input with no assistance from any online resources, while Automated testing requires a primary use of online tools and scripts. Automated testing is more reliable and accurate than Manual testing, however human observation is needed to measure the product's efficiency and user friendliness (apica.io, 2022).

By creating test cases in Visual Studio Code and running 'pytest' in the Git Bash terminal, I have performed Unit (Automated) Testing on my application to test the smallest individual parts of the application and confirm that it works as intended (www.techtarget.com, 2023). In addition, I made use of Jupyter Notebook for testing and debugging the code while implementing the functionality. In order to carry out Automated Testing in VS Code, I was required to run the command 'pip install pytest' while the virtual environment was activated to ensure that the proper software needed for the testing was installed. Next was to create a configuration file in the root folder named 'pytest.ini' to establish which Django settings were to be used, as well as create files for which to define the test cases, i.e. 'test_urls.py' and 'test_views.py'. After these steps were complete, I ran 'pytest' in the Git Bash terminal and got the result seen below.

```

(virt)
seanj@LAPTOP-5CSD6V25 MINGW64 /c/djangowebpage/homepage (main)
$ pytest
===== test session starts =====
platform win32 -- Python 3.11.0, pytest-7.3.1, pluggy-1.0.0
django: settings: homepage.settings (from ini)
rootdir: C:\djangowebpage\homepage
configfile: pytest.ini
plugins: Faker-12.0.1, cov-4.0.0, django-4.5.2
collected 2 items

webpage\tests\test_views.py . [ 50%]
webpage\tests\test_urls.py . [100%]

===== 2 passed in 0.66s =====

```

The screenshot shows a Visual Studio Code interface. The terminal window at the top displays the output of a pytest command, showing that two tests passed in 0.66 seconds. The code editor below shows the source code for test_views.py and test_urls.py. The test_views.py file contains a TestViews class with a test_phish1_get method that uses Django's Client to request a phish1 URL and asserts that the status code is 200 and that the template used is 'phish1.html'. The test_urls.py file contains a TestUrls class with a test_phish1_get method that uses Django's Client to request a phish1 URL and asserts that the status code is 200 and that the template used is 'phish1.html'.

Below is a grid detailing the results of Manual Testing in the Browser:

Function	Description	Expected Outcome	Test Outcome	Test Result (Pass/Fail)	Notes
Navbar Homepage Link Test	Access the Homepage, Guidebook, Recommendations and Login/Logout page, when outside of a Phishing Game?	The user accesses the Homepage from the Navbar link.	The user accessed the Homepage from the Navbar link.	Pass	The Navbar is excluded from the Phishing Games so that the user is less likely to accidentally exit during a session.
Navbar Guidebook Link Test	Access the Guidebook page when outside of a Phishing Game.	The user accesses the Guidebook from the Navbar link.	The user accessed the Guidebook from the Navbar link.	Pass	Text from Guidebook page was originally planned to appear in Phishing Games.
Navbar Recommendations Link Test	Access the Recommendations page when outside of a Phishing Game.	The user accesses the Recommendations from the Navbar link.	The user accessed the Recommendations from the Navbar link.	Pass	The Navbar is excluded from the Phishing Games so that the user is less likely to accidentally exit during a session.
Navbar Login/Logout Link Test	Access the Login/Logout page when outside of a Phishing Game.	The user accesses the Login/Logout page from the Navbar link.	The user accessed the Login/Logout page from the Navbar link.	Pass	Logging in is not required to access application functionality.
Phishing Game 1 Link Test	The link to the 1st Phishing Game on the Homepage?	Sends the user to the 1st Phishing Game.	Sends the user to the 1st Phishing Game.	Pass	An 'Exit' button is present in each Phishing Game to easily return to the Homepage.
Phishing Game 2 Link Test	The link to the 2nd Phishing Game on the Homepage?	Sends the user to the 2nd Phishing Game.	Sends the user to the 2nd Phishing Game.	Pass	An 'Exit' button is present in each Phishing Game to easily

					return to the Homepage.
Phishing Game 3 Link Test	The link to the 3rd Phishing Game on the Homepage?	Sends the user to the 3rd Phishing Game.	Sends the user to the 3rd Phishing Game.	Pass	An 'Exit' button is present in each Phishing Game to easily return to the Homepage.
Checkbox Test	Phishing Game 1 selected checkboxes.	The user remains on the same page if any 'wrong' boxes are selected.	The user remained on the same page because answers with a 'wrong' value were selected.	Pass	If all boxes are selected, both 'right' and 'wrong', the user still remains on the same page.
Radio Button Test	Phishing Game 2 Multiple choice radio buttons.	The application displays the number of correct answers at the top of the page.	The application displayed the number of correct answers at the top of the page.	Pass	If no buttons are selected, the application returns a message letting the user know they must not leave any unattempted.
Audio File Test	Phishing Game 3 audio files.	The audio files play what was originally recorded.	The audio files played what was originally recorded.	Pass	More than one audio file can be played at once without causing issues.

2.6. Evaluation

Currently, this application has not been hosted so there is no data to be used for evaluation. This would be for further research.

3.0 Conclusions

Advantages:

- ➔ The application is easy to use, navigate and understand.
- ➔ It is free to use.

Disadvantages:

- ➔ There are certain, more complex variants of Phishing attacks, e.g. Search Engine Phishing (trendmicro.com) and HTTPS Phishing (bluevouant.com), that are not

the focus of any of the 3 Phishing Games and are simply mentioned as questions within Phishing Game 2.

Strengths:

- ➔ The application provides education on how to counter a prominent cyber threat, as well as direction to sources that describe or provide further protection against said threat.
- ➔ It is not over-saturated with content, making it hard to navigate and find specific functionality within the application.

Limitations:

- ➔ I originally planned for the application to feature a score functionality for each Phishing Game, where the user would receive a higher score based on how many questions were correctly answered. I was unfortunately unable to incorporate this as I was unable to establish a database for the application before the deadline.
- ➔ It was also planned that each of the Phishing Games would have text appear in the bottom right during a session to assist the user. This text would have come directly from the Phishing Guidebook page. While I did not achieve exactly what I envisioned for this, when the user answers incorrectly in Phishing Game 1, some text will appear above the checkboxes giving hints to the user on what to focus on in the email or text message.
- ➔ The same questions are displayed in the same order each time the user starts a session in each Phishing Game. Once the user is able to successfully identify the answers to each question, the application has no new information to provide. Together with the lack of a score functionality, this may lessen the replay value of the overall application.

4.0 Further Development or Research

If I decided to continue developing my Phishing application, a potentially helpful strategy of research that would produce a better general understanding of public awareness and experiences regarding Phishing based attacks, would be to organise a survey. This survey would not be particularly large but would involve people who have had experiences with the threat; either by encountering it or as the victims of an actual Phishing attack. All research carried out for the development of the application thus far has exclusively involved online articles and sites, so a survey would be an effective strategy in providing useful data. As discussed above, expanding the number of interactive Phishing Games in order to cover more types of complex Phishing attacks, rather than solely providing the education for such in the form of answering questions, would increase the application's value of education.

5.0 References

- edapp.com (2022) *10 Phishing Training Software*. Available at: <https://www.edapp.com/blog/phishing-training-software> [Accessed 12 May 2022].
- cofense.com (2022) *Phishing Threats to Your Business*. Available at: <https://cofense.com/knowledge-center/phishing-threats/#:~:text=The%20definition%20of%20a%20phishing,a%20trustworthy%20organization%20or%20entity/> [Accessed 12 December 2022].
- devops.com (2022) *Game Developers Shouldn't Overlook Python's Potential*. Available at: <https://devops.com/game-developers-shouldnt-overlook-pythons-potential/> [Accessed 12 December 2022].
- www.upgrad.com (2020) *Data Structures & Algorithm in Python*. Available at: <https://www.upgrad.com/blog/data-structures-algorithm-in-python/#:~:text=Some%20of%20the%20well%2Dknown,Conquer%20algorithm%2C%20and%20Backtracking%20algorithm/> [Accessed 12 December 2022].
- lucidchart.com (2023) *Complete Guide to Waterfall Project Management Methodology*. Available at: <https://www.lucidchart.com/blog/waterfall-project-management-methodology> [Accessed 12 May 2023]
- tutorialspoint.com (2022) *Python – Databases and SQL*. Available at: [https://www.tutorialspoint.com/python_network_programming/python_databases_and_sql.htm#:~:text=The%20Python%20programming%20language%20has,DML\)%20and%20Data%20Query%20Statements/](https://www.tutorialspoint.com/python_network_programming/python_databases_and_sql.htm#:~:text=The%20Python%20programming%20language%20has,DML)%20and%20Data%20Query%20Statements/) [Accessed 17 December 2022].
- bitlyft.com (2023) *Don't Get Hooked by Vishing: How to Spot and Avoid a Vishing Attack*. Available at: <https://www.bitlyft.com/resources/dont-get-hooked-by-vishing-how-to-spot-and-avoid-a-vishing-attack> [Accessed 1 May 2023]
- armorblox.com (2023) *Spear Phishing 101: What It Is and How To Prevent It*. Available at: <https://www.armorblox.com/learn/spear-phishing/#:~:text=Here%20are%20some%20common%20reasons,your%20employer%20that%20look%20legitimate> [Accessed 1 May 2023]
- educative.io (2022) *What is MVT structure in Django?* Available at: <https://www.educative.io/answers/what-is-mvt-structure-in-django> [Accessed 12 May 2023]
- developer.mozilla.org (2023) *Django Introduction – Learn web development*. Available at: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction> [Accessed 13 May 2023]
- apica.io (2022) *Manual VS Automated Testing: An Overview*. Available at: [https://www.apica.io/difference-between-automated-manual-testing/#:~:text=In%20manual%20testing%20\(as%20the,of%20any%20successful%20software%20project/](https://www.apica.io/difference-between-automated-manual-testing/#:~:text=In%20manual%20testing%20(as%20the,of%20any%20successful%20software%20project/) [Accessed 16 December 2022].
- www.techtarget.com (2023) *What is Unit Testing? Definition from WhatIs.com*. Available at: <https://www.techtarget.com/searchsoftwarequality/definition/unit-testing#:~:text=Unit%20testing%20is%20a%20software,independently%20scrutinized%20for%20proper%20operation> [Accessed 1 May 2023]

- trendmicro.com (2022) *What are the Different Types of Phishing?* Available at: https://www.trendmicro.com/en_ie/what-is/phishing/types-of-phishing.html/ [Accessed 12 December 2022].
- bluevoyant.com (2022) *8 Phishing Types and How to Prevent Them.* Available at: <https://www.bluevoyant.com/knowledge-center/8-phishing-types-and-how-to-prevent-them/> [Accessed 12 December 2022].

6.0 Appendices

6.1. Project Proposal



National College of Ireland

Project Proposal

<"Phish Off" – Don't Get Hooked!>

<20/10/2022>

<Bachelor of Science Honours in Computing>

<Cybersecurity>

<2022/2023>

<Seán Mc Donnell >

<x19490822>

<x19490822@student.ncirl.ie>

Contents

1.0	Objectives	28	
2.0	Background	28	
3.0	State of the Art	28	
4.0	Technical Approach	28	
5.0	Technical Details	29	
6.0	Special Resources Required		Error! Bookmark not defined.
7.0	Project Plan	29	

8.0 Testing 31

7.0 Objectives

This Project aims to increase cybersecurity awareness through an interactive application that will inform, educate and help users identify and recognise cybersecurity threats. In particular, there will be a focus on the threat of Phishing. This application will be user friendly for all ages. Many people do not fully recognise what Phishing is and this app will bring more understanding and awareness to the user. Phishing refers to the act of a malicious user impersonating a trustworthy source in order to steal personal information from an individual or company by deploying malware. This type of attack is delivered mostly in the form of emails as well as advertisements in trustworthy websites (cofense.com, 2022).

8.0 Background

As people increasingly engage with digital devices for work, socialising and entertainment, the possibility of having their data stolen or compromised is an ever increasing possibility. During my 3rd year work experience placement in a health facility, there was a proactive approach taken to alert employees to the cybersecurity threat of Phishing. It occurred to me that the general public may not have enough awareness of Phishing. Therefore, I chose this topic for my 4th year specialisation Computer Project.

To meet these objectives, I plan to develop a cybersecurity-based Phishing interactive application that will provide an educative and entertaining experience for learning the 'dos and don'ts' of Phishing. The knowledge and awareness of the various potential threats gained through this application will help users to identify evidence of Phishing in emails, text messages and phone calls. The users will therefore be more cyber aware when using their digital devices and social media.

9.0 State of the Art

In researching similar applications, I discovered several Phishing simulators, e.g. Infosec IQ (Antipov, 2021) and for younger users, Cyber Defense Quiz and Privacy Pirates (Keeley, 2020).

The application I plan to develop will be broken down into different mini-games (i.e. different games within the application), each focusing on a different cybersecurity threat or testing the user's knowledge of what is considered a 'red flag' when interacting with an anonymous user. It will also include a guidebook feature where text appears in the bottom right at certain points, providing the user with facts and advice about Phishing and being cautious in the online environment.

10.0 Technical Approach

Using a specified coding language, I plan to develop the application by designing the different mini-games separately and later on establishing inheritance for them to work as one web application. When the application opens, the user will see the main menu, complete with the title 'Phish-Off', where they can select which mini-game they wish to play by clicking on the provided text. The goal of this project is to provide users with the knowledge of how to identify Phishing threats and as a follow up action, cease play and report if necessary. In my project I plan to include a quiz mini-game to test what they have learnt after experiencing the other 3 mini-games.

The first mini-game has the user receiving emails and checking them for evidence of Phishing. They will be told to answer 'yes' or 'no' when questioned for evidence of Phishing and must answer

correctly 7 consecutive times. The second mini-game teaches on how to avoid being manipulated when interacting with an anonymous user by simulating an online gaming conversation. The player will be asked a question from the anonymous user and using a traffic lights system, they must respond with either green (safe), orange (careful) or red (must avoid). In the third mini-game, the user is asked what to do next after encountering such a threat, e.g. stop, don't click or exit . They will also receive hints from the guidebook in the bottom right.

In order to identify the requirements for successfully developing this project, I will start by researching and designing my Application. I will construct ER Diagrams to create a clear image of the game functionality. A helpful resource will include reading coding projects from past academic years and also my second year team project. Additionally, I will research online sources when implementing the code for the project and reference appropriately. I will divide the project requirements into tasks consisting of the individual actions performed in the app, rather than attempting to develop an entire mini-game at once without a scene of order. After determining what is the most appropriate coding language for programming the different mini-games of the game application, I will then choose an integrated development environment (IDE), e.g. NetBeans, that I feel is appropriate for the development of this app.

11.0 Technical Details

After researching which implementation languages would be the most appropriate for the development for my project, I found that JavaScript and Python are the recommended choices for programming a web application, as said by (W3Schools, n.d.). In addition, Python is compatible with many popular programming libraries such as NumPy, Matplotlib, TensorFlow and Apache Spark, as said by (Meltzer, 2022), and is a popular choice for programming thanks to its easy-to-understand syntax, according to (Kozacz, 2022). It is also not restricted depending on what kind of application is being designed and can be used as per the specific requirements, according to (Sharma, 2020). After considering this, I believe that Python will be my choice of implementation language for developing my Phishing application. I have also researched Python frameworks recommended for developing an interactive application, with PyGame being the most popular option due to being used on a variety of platforms and being easy to use. It also works alongside the cross platform SDL library (www.unikaksha.com, 2022).

When playing a mini-game within my proposed application, the experience will not be the same each time, i.e. I intend questions to be asked in a different order between sessions as well as not having the exact same questions appearing each time. Python supports the Sort algorithm which can be used to set the order of items and with the shuffle() method, the items will appear in a different order each time the game is started up. The guidebook mentioned earlier can be explored fully by clicking the appropriate text on the main menu, and the user can search for the tip they need to learn about Phishing and the other previously mentioned threats. Python supports the Search algorithm which can be used to create an easily accessible search bar when accessing the guidebook. The traffic light system featured in the second mini-game has the user respond to a question with either green, orange or red. This response could be entered in a text box that appears below the question or by clicking on one of three buttons corresponding to the colour, achieved by implementing a GUI. I plan to customise the application with colours and images that represent the Sci-Fi theme mentioned earlier. Python's open() function enables you to open and print the contents of a file or image onto a web page.

12.0 Project Plan

The project work begins by confirming the programming language I will utilise, as well as appropriate libraries, algorithms, mapping out the functionality of the project with diagrams and having frequent and productive interactions with my assigned supervisor. I have listed down the requirements, tasks and research topics on an Excel sheet to keep record and serve as a checklist during the development of my application. I believe it will be more productive to start by designing the three mini-games before working on the menu, guidebook and aesthetics theme, since they are the most important part and will require the most effort and attention, but before I begin developing the code, I must visualise the functionality on ER diagrams and identify the stages of development. From experience of developing applications for assignments in previous academic years, the strategies of breaking the project requirements down into tasks and visualising the how it will look through diagrams are effective at simplifying a large project such as this. By the end of semester 1, I intend to have regular contact with my supervisor established, at least twice weekly, and to have a structure designed for the three mini-games. I do not expect to have the code for such fully developed by then, and I am expecting to see changes to my approach as the year progresses, which may include the discarding of some intended features. I am also planning to build on the concept that I have currently proposed.

The following is an outline of the project plan:

- **Phase 1 (Resource Gathering):**

Make a list of libraries, algorithms, languages and additional tools will be needed to design the application.

- **Phase 2 (Diagrams & Research):**

Visualise how the application will operate using ER Diagrams and research tutorials on the development of similar applications.

- **Phase 3 (Q&A with Supervisor):**

Establish regular times for online and/or face to face meetings with supervisor to discuss the progress of development and any changes that are needed.

- **Phase 4 (Designing the Application):**

Begin inputting the code and algorithms with a clear image of how the application will function. Start by designing the 3 mini-games.

- **Phase 5 (Mid Point Implementation):**

Prepare video presentation, documentation and slides to show current progression of the application.

- **Phase 6 (Continue Design & Establish Inheritance):**

Continue working on designing the different mini-games and insert algorithms to allow them to work as one application.

- **Phase 7 (Creating the Aesthetics):**

Add the background images as well as the starting screen and guidebook feature.

- **Phase 8 (Testing the System):**

Carry out software tests on the system to ensure that it is functioning at the best possible capacity.

- **Phase 9 (Final Implementation):**

Prepare video presentation and documentation along with the submission of the finished application.

A Gantt Chart demonstrating planned schedule for project.

	Oct-22	Nov-22	Dec-22	Jan-23	Feb-23	Mar-23	Apr-23	May-23
Phase 1								
Phase 2								
Phase 3								
Phase 4								
Phase 5								
Phase 6								
Phase 7								
Phase 8								
Phase 9								

13.0 Testing

As I am considering developing the different mini-games of my project individually and having them operate as part of one application, I will use an Integration test to evaluate the functionality and ensure that they will run without issue. Another useful test to consider is a Unit test where individual pieces (or units) of code are tested for issues which can be easily and quickly amended in this procedure. In contrast, issues within an Integration test are harder to identify and take longer to fix, but both tests are considered equally helpful in software planning. An Integration test is also normally carried out after any Unit testing, (PractiTest.com, 2022). Due to the different speeds and situations the tests are used for, it is recommended to separate Unit and Integration tests into different folders (Ostapczuk, 2022). System testing is where all components of the application are linked together to confirm that it operates as planned. Both Integration and System tests do not require access to internal system components such as code, (Nanda, 2022).

Unit testing is mandatory for the development of any project that involves coding, regardless of which language is involved, so I will definitely need it for mine. Python libraries recommended for implementing a Unit test are PyUnit and PyTest (Chng, 2022). System testing is done on the fully integrated application as well as any external resources, meaning that it is done at the end or close to the end of development (Hamilton, 2022). This means that regardless of whether or not a System test will be suitable for my project, I will research exactly what needs to be incorporated into it and determine with my supervisor if it is needed. As stated above, an Integration test will be quite suitable for my application.

References

- Antipov, A., 2021. *Top Nine Phishing Simulators*. [Online]
 Available at: <https://resources.infosecinstitute.com/topic/top-9-free-phishing-simulators/>
 [Accessed 30 October 2022].
- Chng, Z. M., 2022. *A Gentle Introduction to Unit Testing in Python*. [Online]
 Available at: <https://machinelearningmastery.com/a-gentle-introduction-to-unit-testing-in-python/#:~:text=Unit%20testing%20is%20a%20method,works%20correctly%20and%20as%20intend>

ed.

[Accessed 30 October 2022].

cofense.com, 2022. *Phishing Threats to Your Business*. [Online]

Available at: <https://cofense.com/knowledge-center/phishing-threats/#:~:text=The%20definition%20of%20a%20phishing,a%20trustworthy%20organization%20or%20entity.>

[Accessed 30 October 2022].

Hamilton, T., 2022. *What is System Testing?*. [Online]

Available at: <https://www.guru99.com/system-testing.html#:~:text=System%20Testing%20involves%20testing%20the,End%20to%20End%20testing%20scenario.>

[Accessed 30 October 2022].

Keeley, J., 2020. *7 Internet Safety Games*. [Online]

Available at: <https://www.makeuseof.com/tag/6-internet-safety-games-kids-cyber-smart/>

[Accessed 30 October 2022].

Kozacz, H., 2022. *Game Developers Shouldn't Overlook Python's Potential*. [Online]

Available at: <https://devops.com/game-developers-shouldnt-overlook-pythons-potential/>

[Accessed 26 October 2022].

Meltzer, R., 2022. *What is a Programming Library?*. [Online]

Available at: <https://careerfoundry.com/en/blog/web-development/programming-library-guide/#:~:text=A%20programming%20library%20is%20a,few%20different%20pre%2Dcoded%20components.>

[Accessed 26 October 2022].

Nanda, V., 2022. *What is System Testing*. [Online]

Available at: <https://www.tutorialspoint.com/what-is-system-testing-definition-types-examples/#:~:text=Example%20of%20System%20Testing&text=Each%20component%20of%20the%20automobile,is%20known%20as%20unit%20testing.>

[Accessed 29 October 2022].

Ostapczuk, P., 2022. *Separate Gradle Tasks for Unit and Integration Tests*. [Online]

Available at: <https://inspeerity.com/blog/separate-gradle-tasks-for-unit-and-integration-tests>

[Accessed 30 October 2022].

PractiTest.com, 2022. *Unit Test vs Integration Test*. [Online]

Available at: <https://www.practitest.com/qa-learningcenter/resources/unit-test-vs-integration-test/#:~:text=Unit%20Testing%20is%20a%20kind,interactions%20and%20interfaces%20between%20modules.>

[Accessed 29 October 2022].

Sharma, R., 2020. *Data Structures & Algorithm in Python*. [Online]

Available at: <https://www.upgrad.com/blog/data-structures-algorithm-in-python/#:~:text=Some%20of%20the%20well%2Dknown,Conquer%20algorithm%2C%20and%20Backtracking%20algorithm.>

[Accessed 26 October 2022].

W3Schools, n.d. *Python Tutorial*. [Online]

Available at: <https://www.w3schools.com/python/default.asp>

[Accessed 26 October 2022].

www.unikaksha.com, 2022. *Top 6 Python Framework for Game Development*. [Online]

Available at: [https://www.unikaksha.com/top-6-python-framework-for-game-](https://www.unikaksha.com/top-6-python-framework-for-game-development/#:~:text=Pygame%20is%20the%20most%20popular,mouse%2C%20keyboard%2C%20a)

[development/#:~:text=Pygame%20is%20the%20most%20popular,mouse%2C%20keyboard%2C%20and%20joystick](https://www.unikaksha.com/top-6-python-framework-for-game-development/#:~:text=Pygame%20is%20the%20most%20popular,mouse%2C%20keyboard%2C%20and%20joystick).

[Accessed 26 October 2022].

13.1. Reflective Journals

Supervision & Reflection Template

Student Name	Seán Mc Donnell
Student Number	x19490822
Course	BSc Honours in Computing
Supervisor	Vanessa Ayala-Rivera

Month: October

What?

I spent time brainstorming and researching possible project ideas and looked at examples of past projects from previous 4th year students. I narrowed my choices down to 2 topics, Password Management and Phishing. I decided on Phishing as I feel many people are not aware of it and education is needed in this area. I recorded the Project Pitch video and completed the Project Proposal. I have found it challenging trying to decide on a suitable topic and trying to find a particular angle that would set my project apart from existing original pieces.

So What?

- I feel it took some time to decide on an idea and then get started on trying to define that idea.
- I enjoyed doing the elevator pitch video.
- I am still struggling to find that unique angle for my project proposal. I think defining it and ensuring that the technical details are correct are challenging and will continue to be challenging until I get more feedback from my supervisor.

Now What?

- I need to continue researching this topic in as much detail as possible. I need to be very clear about existing products similar to what I am proposing.
- I will work closely with my supervisor on all the details required.
- I have been given additional resource support for the project which will be very beneficial.

Student Signature

Seán Mc Donnell

Supervision & Reflection Template

Student Name	Seán Mc Donnell
Student Number	x19490822
Course	BSc Honours in Computing
Supervisor	Vanessa Ayala-Rivera

Month: November

What?

I spoke with my supervisor and established a regular weekly meeting time. During our meetings we discussed the following issues:

- After reviewing the pitch video, recommendations were suggested which included developing an application that is more generalisable in terms of age group.
- Another recommendation was to reduce the gamification content and to avoid sensitive ethical issues regarding neurodivergent groups.
- I documented the functional and non-functional requirements of my application as well as a UML class diagram to provide my supervisor with a clear understanding of how the application operates.

I have had additional support classes with Emer Thornbury who recommended certain tasks and keeping lists and timelines of what needs to be accomplished, e.g. which frameworks would pair well with my programming language of choice as well as the size and type of database used to save the data.

So What?

- Since I've had to balance working on the Computing Project with the CAs from my other modules, I haven't been able to implement much of any code and have been mostly researching, planning and mapping out the application's functionality and software requirements.
- I've made it so that the application is now focused towards a more generalised age group.
- There are still some elements of gamification within my application, but as it stands, it is so that there is data to be saved when it is in use.
- I am concerned that my application is lacking a unique angle to set it apart from similar projects since I have removed the autism friendly approach.

Now What?

- I need to continue researching this topic in as much detail as possible. I need to be very clear about existing products similar to what I am proposing.
- I will work closely with my supervisor on all the details required.
- I will make full use of the additional resource support that has been provided to me.

Student Signature



Supervision & Reflection Template	
----------------------------------------------	--

Student Name	Seán Mc Donnell
Student Number	x19490822
Course	BSc Honours in Computing

Month: December

What?

I submitted the Mid-Point Presentation for my project, which included a Technical Report, PowerPoint slides, a video presentation and the code for the prototype. I designed a detailed Use Case Diagram to include in the Report and provided a description of the requirements for 3 important Use Cases for the 3 interactive Phishing Games. Additionally in the Report, I provided visual representations of how the interactive games are planned to look once the application is functioning as intended. For the Prototype, I managed to design a working Homepage and 2 additional pages for a Phishing Guidebook and Recommended Reading when the application is running on the Browser, after creating a virtual environment for it using the Python framework Django. I pasted the links to the video and the prototype code to a txt file and placed it as well as the Report and PowerPoint slides into a zipped folder before submitting it to Moodle. I speak with my supervisor once a week to discuss any issues arising and to work on feedback received at these meetings.

So What?

- There are still some elements of gamification within my application, i.e. a score, but it serves the purpose of reflecting the user's performance, therefore I feel it is justified to include it.
- I am still working on developing my project idea and having an angle that reflects some innovation.
- The application was originally intended to cover a variety of cyber threats in addition to Phishing, but for the sake of avoiding overcomplication, it now focuses exclusively on the one cyber threat.
- I only had ideas for 2 Phishing Games prior to starting the Technical Report, but after researching the various types of Phishing Attacks, I decided on 3 ideas and I believe it will be beneficial to add at least 1 more.
- In completing the Technical Report, I gained a better understanding of what I need to accomplish in order to successfully develop my application and what I want the application to be.

Now What?

- I need to continue researching my topic in as much detail as possible. I need to be very clear about existing products similar to what I am proposing.
- I will continue to work closely with my supervisor on all the details required.
- I have been given additional resource support for the project which has been very beneficial.

Student Signature	<i>Seán Mc Donnell</i>
--------------------------	------------------------

Supervision & Reflection Template

Student Name	Seán Mc Donnell
Student Number	x19490822
Course	BSc Honours in Computing
Supervisor	Vanessa Ayala-Rivera

Month: January

What?	<p>I have expanded the functionality of my application since submitting the prototype with the Mid-Point Presentation. I have added more information to the Phishing Guidebook page and added links to the Recommendations page. These links direct the user to sites detailing the HSE 2021 and NHS 2022 attacks where Phishing was involved in the execution of both, as well as a link to a list of anti-Phishing software to install on your devices. I have begun developing the first Phishing Game, which involves emails and text messages, starting with implementing an image of a Phishing email. I have made sure to get in touch with my supervisor to set up a regular meeting time for this semester.</p>
So What?	<ul style="list-style-type: none"> • In Phishing Game 1, the image of the message is intended to have several toggleable buttons that are used to identify the traits of Phishing. I have yet to successfully implement the buttons but I have begun working on the code for it. • I planned to include at least six questions for this first Phishing Game, but I feel that timewise I should divide my effort between all the elements of the application's functionality rather than spending too much time focusing on a single section. • I had also planned to have text from the Phishing Guidebook appear whilst interacting with the application, but now I feel that this is not a priority and I'm going to work on implementing the main functions for the application before getting to the guidebook help feature.
Now What?	<ul style="list-style-type: none"> • I will continue to work closely with my supervisor on all the details required. • I have been given additional resource support for the project which has been very beneficial.
Student Signature	<i>Seán Mc Donnell</i>

Month: February

What?

I have continued to develop Phishing Game 1, implementing different pages for different questions on malicious emails and text messages. I have also begun developing Phishing Games 2 and 3, the former providing questions in the style of a multiple choice quiz and the latter asking the user to play back the provided audio file and identify whether or not it is a Vishing (a category of phishing) call. I originally planned for Phishing Game 1 to feature buttons placed on the email image, but after discussing with my supervisor, I decided to change it to checkboxes. Additionally, while I have managed to implement a link between questions/pages in Game 1, my intention is that you can only move to the next question when you have answered the current question correctly.

So What?

- In Phishing Game 1, the intention is that the user is only able to progress to the next question if they have ticked the appropriate boxes upon hitting 'Next Page'. Currently, clicking this button will take you to the next page even if no boxes are ticked.
- A database for my application must also be implemented to store the data saved when interacting with the Phishing Games.

Now What?

- I will continue to work closely with my supervisor on all the details required.
- I have been given additional resource support for the project which has been very beneficial.

Student Signature



Supervision & Reflection Template


Student Name	Seán Mc Donnell
Student Number	x19490822
Course	BSc Honours in Computing
Supervisor	Vanessa Ayala-Rivera

Month: March

What?

I began implementing a database for my phishing application this month, creating models and forms in Django meant to store the data processed when the user is interacting with the application, i.e. their answers to the questions.

So What?

<ul style="list-style-type: none"> • In Phishing Game 1, the submit button at the bottom of the page still takes the user to the next question even if nothing is ticked. • I have discussed with my supervisor that I should record my own audio samples for Phishing Game 3 as it would be hard to find an audio file online that's exactly what I'm looking for. 	
Now What? <ul style="list-style-type: none"> • I will continue to work closely with my supervisor on all the details required. • I have been given additional resource support for the project which has been very beneficial. • I will research examples of Vishing calls and record my own in a way I envisioned for Phishing Game 3. 	
Student Signature	

Supervision & Reflection Template

Student Name	Seán Mc Donnell
Student Number	x19490822
Course	BSc Honours in Computing
Supervisor	Vanessa Ayala-Rivera

Month: April

What? <p>I managed to work out the functionality for Phishing Game 1, allowing the application to only move to the next question if the correct boxes are ticked for the current question. After discussing with my supervisor, I decided that it isn't necessary to set up a database to get the application working the way I intended.</p> <p>Audio files have also been incorporated into Phishing Game 3, where the user must listen to each and determine if it is a Vishing call or not by simply clicking either the Yes or No radio buttons. I have also begun preparing to perform Unit Testing once I am satisfied with the level of functionality and I am filling in my final report with all the changes made since the Mid-Point submission.</p>
So What? <ul style="list-style-type: none"> • I do not believe that I will be able to get the guidebook text to appear during a Phishing Game before the final submission. • I planned to include at least one additional Phishing Game before the end if there was time, but I must focus on developing the existing ones at this point in time.
Now What?

- I intend to include more background images and edit the text font to make the application appear more visually pleasing.
- I have to work out how to get Phishing Games 2 and 3 to display the exact number of correct answers when the user hits the submit button.
- For the report, I plan to redraw GUI images using online tools, i.e. Draw.io, as apposed to drawing them out by hand.

Student Signature

Sean Mc Donnell