

National College of Ireland

Bachelor of Science (Honours) in Computing

Cybersecurity Specialisation

Year 2022/23

Nathan Savage

18395306

x18395306@student.ncirl.ie

The Snooper (Vulnerability Intelligence Application) Technical Report

Contents

Executive Summary.....	2
1.0 Introduction	2
1.1. Background	2
1.2. Aims.....	2
1.3. Technology.....	2
1.4. Structure	3
2.0 System.....	3
2.1. Requirements.....	3
2.1.1. Functional Requirements.....	3
2.1.1.1. Use Case Diagram	3
2.1.1.2. Requirement 1: Vulnerability Identifier	4
2.1.1.3. Requirement 2: Log In.....	5
2.1.1.4. Requirement 3: Scoring System	7
2.1.1.5. Requirement 4: Profile Page	9
2.1.2. Data Requirements	11
2.1.3. User Requirements	11
2.1.4. Usability Requirements.....	11
2.2. Design & Architecture.....	12
2.3. Implementation	12
2.4. Graphical User Interface (GUI).....	20
2.5. Testing.....	24
2.6. Evaluation.....	31
3.0 Conclusions	31
4.0 Further Development or Research	32
5.0 References	33
6.0 Appendices.....	34
6.1. Project Proposal	34
6.2. Reflective Journals	39

Executive Summary

This report is on a vulnerability intelligence application that I am developing for my final project. It is designed to give users an experience of vulnerability intelligence, by compiling data on vulnerabilities that they are under threat from and provide some analysis on what could be more dangerous. The program is made using HTML, CSS, Python and MySQL as the key technologies. It was developed using the Visual Studio Code IDE. The project has features including a search function, a profile page with editable information, a scoring system, and a PDF generator. These features have been tested and checked to ensure they work with results posted below. The project had many difficulties, these include the first iteration of the code being shelved and work concluding on it. The code was restarted and the finished product that has been submitted is the second iteration. The code had to be restarted due to poor coding choices and a very ambitious plan for the make up of the website and the code of the application. For more information on this, I have described it more in the conclusions of the report. I previously explained my project as a scanner which can be seen in my proposal, however it is a vulnerability intelligence application that searches for the CVEs.

1.0 Introduction

1.1. Background

The idea for a project like this came after thinking back on my internship in Irish Life. While on my internship, I was given duties that related to the day-to-day business and also some tasks that related to monthly patching. Irish Life is a big company that is part of an even larger global company, I wondered how they determined what needed to be done for the monthly patching, realising the amount of people that are involved from North America, Canada to Europe. I then began to think of how small to medium size businesses decide on what they patch to ensure the business continues to run smoothly. A big part of Cybersecurity is vulnerabilities that can impact a business and an IT company, so I decided I wanted to attempt to create an application for small/medium size businesses that can identify vulnerabilities that can attack a business's data and assets.

1.2. Aims

I want to provide vulnerability intelligence to small/medium size businesses to understand the threats that they need to be aware of and have some information that will let them prioritise what needs to be patched.

1.3. Technology

For this project, I will be using the Visual Studio Code IDE to create it and I will be creating the front-end web page using HTML and CSS that will provide users a place to identify known vulnerabilities that can affect their systems and give them information and scores on each vulnerability so they can decide what to patch and when. The backend code will be made in Python, this is where the functions and features of the project will be coded and stored. The databases aspect will be done using MySQL and this will hold the user's data and any data that they decide to send on to the application such as applications and software they utilise.

1.4. Structure

This document is made up of six sections with the first being the introduction, which this section is a part of. The next section is the system section which gives a further breakdown of the requirements of the application, this is shown in use cases, explaining the design and architecture of the application, how it is being implemented, the UI, its testing strategies, and the evaluation of the application. The next section is the conclusions where the application and project as a whole is dissected for its advantages, disadvantages, strengths, and limitations. The fourth section is the further development or research section where the future possibilities of the project could go. Finally, there is the references section and the appendices sections, the references are self-explanatory, being where any references are put, while the appendices will contain all the other documents that I have been working on for this project, such as the proposal and monthly journals.

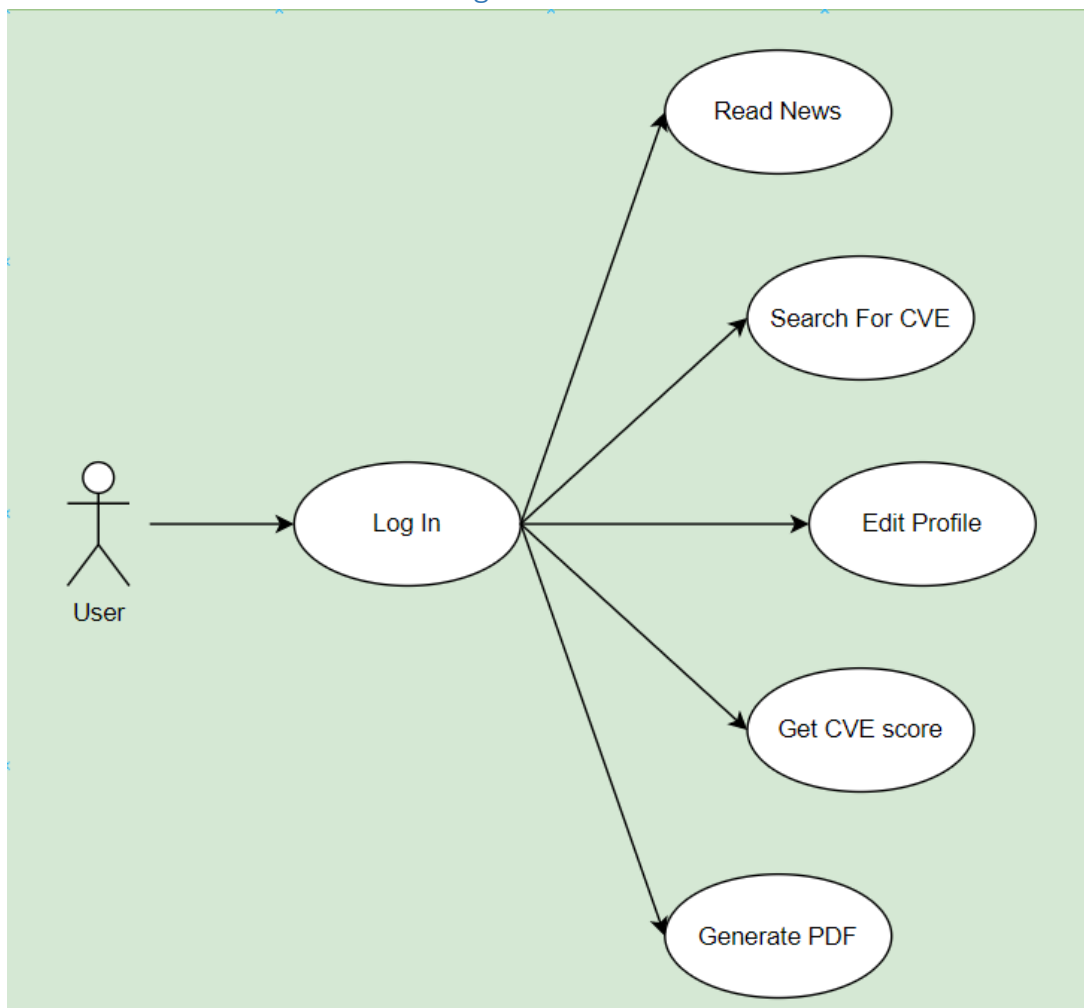
2.0 System

2.1. Requirements

2.1.1. Functional Requirements

There are a few requirements that this project needs to function. These functions range from logging in to searching for different CVEs that adhere to a given key word. This project is made with the idea that it will serve multiple users looking for different CVEs.

2.1.1.1. Use Case Diagram



2.1.1.2. Requirement 1: Vulnerability Identifier

2.1.1.2.1. Description & Priority

The first requirement I will be discussing is the main feature of this application, I want to be able to search the CVE Mitre database for CVEs and be able to display the results on my web page. To do this I utilised the BeautifulSoup web scraper to get the key word, search it with the given URL and take the results then to display them in a table showing the CVE ID and the description for the CVE.

2.1.1.2.2. Use Case

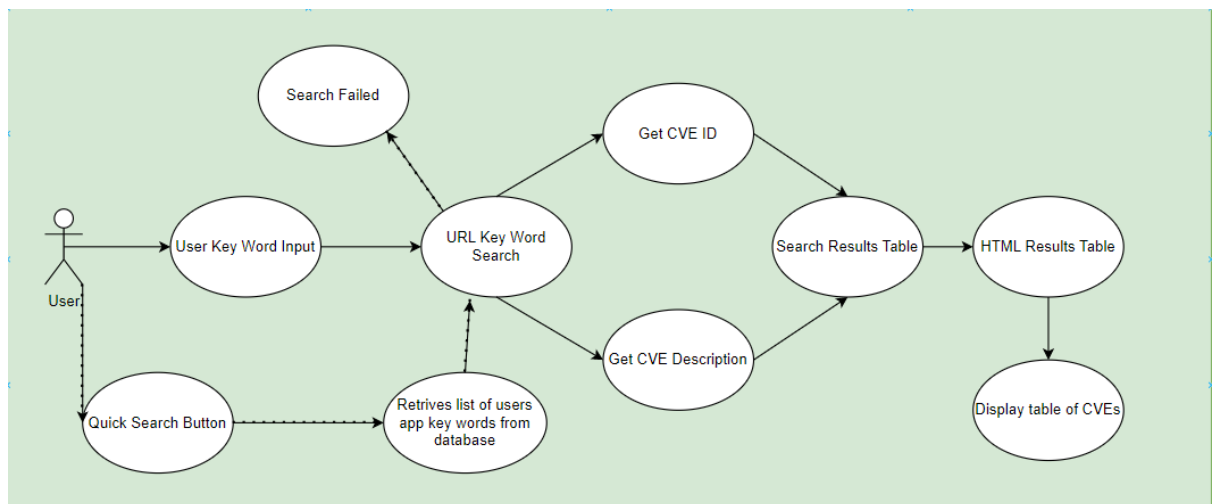
Scope

This Use Case is to show the search feature of the project.

Description

The search feature of the project is coded into the Python and finds CVEs for the user when given a specific key word. The results it finds are the CVE ID and its description these are then inputted into a table and returned to the user on the HTML page.

Use Case Diagram



Flow Description

Precondition

The application is fully functional with a user logged.

Activation

The Use Case starts when the user presses the run button after inputting a key word to search.

Main flow

1. The actor inputs a key word.
2. The code inputs the key word into the URL for the search.
3. The web scraper takes the CVE ID and description that appear after the search is successful.
4. It then puts them into a results table.
5. This is then sent to the HTML code.
6. It is displayed and returned to the user to view.

Alternate flow

A1: Search Failed

1. The actor inputs a key word.
2. The code inputs the key word into the URL for the search.
3. The search fails due to a failed connection.

A2: Quick Search

1. The actor presses the Quick Search button.
2. Code calls for the user's apps from the database.
3. The database returns the list of apps inputted by the user.
4. The list is inputted to the URL for the search.
5. The web scraper takes the CVE ID and description that appear after the search is successful.
6. It then puts them into a results table.
7. This is then sent to the HTML code.
8. It is displayed and returned to the user to view.

Termination

The application will provide a list of CVEs in a table that have mentioned the user key word search. The table is shown on the HTML page for the user.

Post condition

The application waits for the next use.

2.1.1.3. Requirement 2: Log In

2.1.1.3.1. Description & Priority

The next requirement is the log in screen, this is a very important part of the application as this will allow the user to access the application with their specific details.

2.1.1.3.2. Use Case

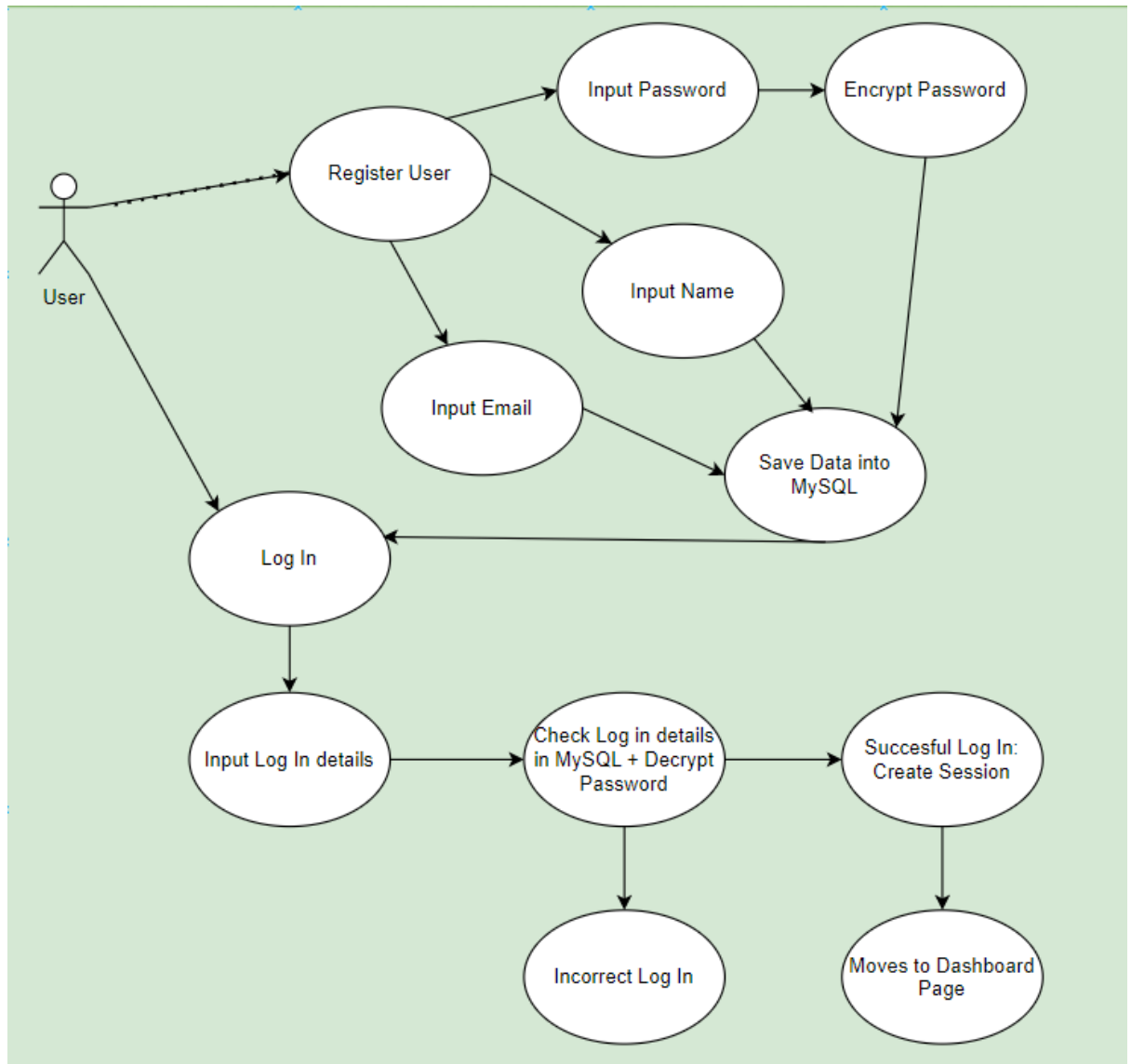
Scope

This Use Case is to show the log in feature of the project.

Description

The log in feature is seen through the HTML of the project with the script of the log-in feature being in the Python. The password is encrypted so it must be decrypted to be read in the database for the log-in to be successful. Once that is done the user can access the web application.

Use Case Diagram



Flow Description

Precondition

The application is fully functional and running and the user has an account.

Activation

The Use Case starts when the user opens the page to access the web application.

Main flow

1. The system opens and waits for an input.
2. The actor inputs their log in info.
3. The system checks the log in info and decrypts the password.
4. The log in is successful and a session is created to track the authentication of the user on the web application.
5. The user is sent to the dashboard screen.

Alternate flow

A1: Incorrect Log In

1. The system opens and waits for an input.
2. The actor inputs their log in info.
3. The system checks the log in info and decrypts the password.
4. The info inputted is incorrect and the user is refused access.
5. The log in page is reloaded for the user to try again.

A2: Register Account

1. The system opens and waits for an input.
2. The user presses the register button.
3. The system opens the registration page.
4. User inputs name, email address and a password.
5. Password is encrypted.
6. Account information is saved to the MySQL database.
7. User is sent to the log in page.

Termination

The system will move the user onto the main dashboard page for the user to interact with the application.

Post condition

The application waits for the next use.

2.1.1.4. Requirement 3: Scoring System

2.1.1.4.1. Description & Priority

The next requirement is the scoring system and the PDF generator for the CVEs. This is in response to the user performing a search and choosing a CVE they wish to get more information about and pressing the score button beside the CVE on the dashboard page.

2.1.1.4.2. Use Case

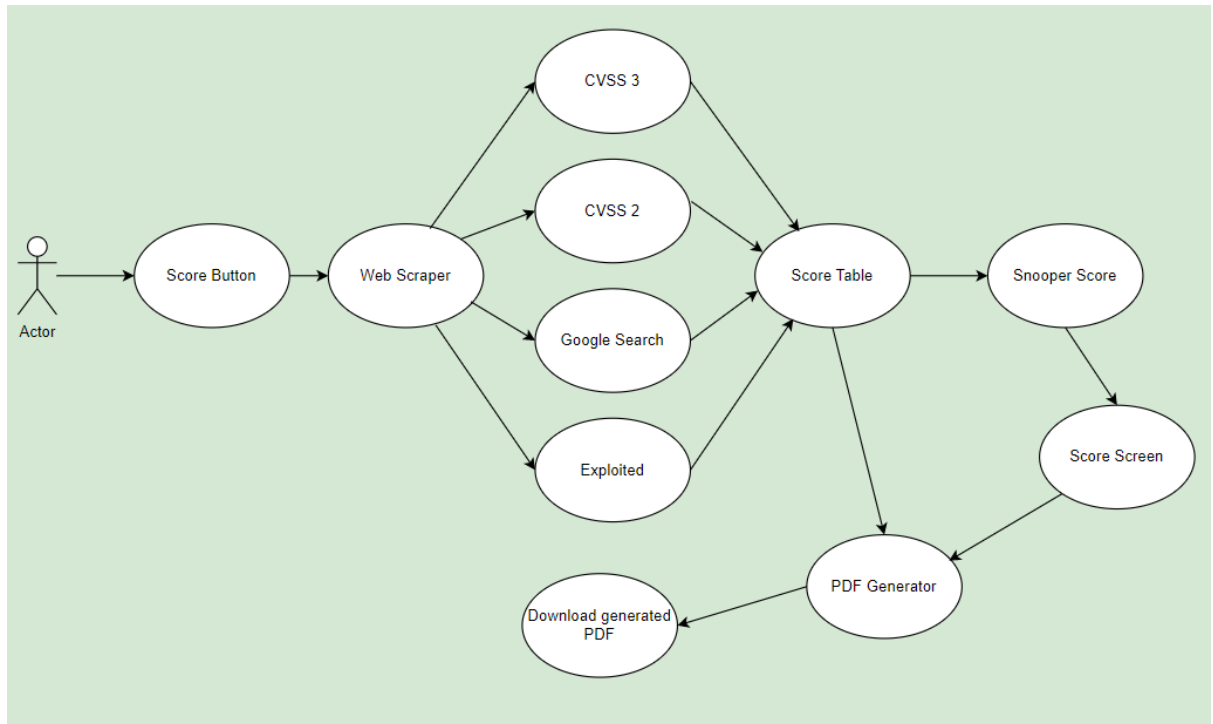
Scope

This Use Case is to show the scoring feature and PDF generation of the project.

Description

The scoring feature is a major part of the application, as it gives the user the most amount of information on CVEs. This part gives them the information on what CVE's are the most important to combat and ensure their systems are safe.

Use Case Diagram



Flow Description

Precondition

The user is logged in and has performed a search function.

Activation

The Use Case starts when the user presses the score button on the dashboard page.

Main flow

1. The user performed a search.
2. The user presses the score button on the dashboard page.
3. The system begins to web scrape the relevant information relating to the CVE that was searched for.
4. The web scraper gets information for CVSS 3, CVSS 2, Google Search and Exploited.
5. The information found is put into a table on the score screen.
6. The information is turned into a snooper score.
7. The table and score are displayed on the Score page which the user is brought to.

Alternate flow

A1: Generate PDF

1. The user gets to the score screen with the table and information loaded.
2. Presses PDF generator button.
3. Code takes the table and inputs it onto a PDF for the user.
4. PDF is downloaded onto the user's device.

Termination

The system will move the user onto the score page.

Post condition

The application waits for the next use.

2.1.1.5. Requirement 4: Profile Page

2.1.1.5.1. Description & Priority

The next requirement is the profile page and the input of information on that page. This page shows the user their information they have inputted into the database. It gives them an option to edit this information and it also allows them to input any applications they use to the database for use in the quick search function.

2.1.1.5.2. Use Case

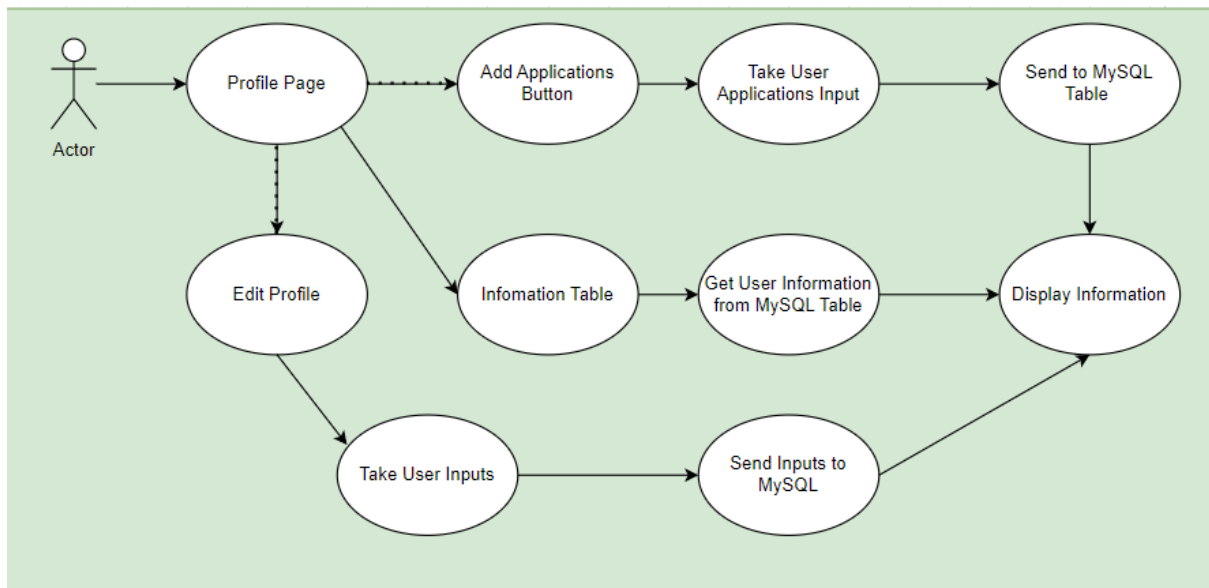
Scope

This Use Case is to show the profile page with the edit feature and the applications used feature.

Description

The profile page is the personal page where each user is able to view and edit their own information, this can help with changing email addresses and users name in cases where it is a company run account. There is also a feature on this page that allows users to input applications and software they use as part of their work, this allows users to use the quick search function.

Use Case Diagram



Flow Description

Precondition

The user is on the profile page.

Activation

The Use Case starts when the user presses the button to access the profile page in the nav bar.

Main flow

1. The user accessed the profile page.
2. The system loads the page and the information for it.
3. The system communicates with the database to send the relevant user information.
4. The system displays the information for the user to view.

Alternate flow

A1: Edit Profile

1. The user opens the profile page, and it loads up.
2. The user presses the edit profile button.
3. User inputs new profile information.
4. The system sends the data to the database and updates it.
5. User is showed the profile page with updated information.

A2: Edit Profile

1. The user opens the profile page, and it loads up.
2. The user presses the add applications button.
3. User inputs new application information.
4. The system sends the data to the database and inserts it into the relevant table.
5. User is showed the profile page with updated application table information.

Termination

The system will move the user to the profile page.

Post condition

The application waits for the next use.

2.1.2. Data Requirements

Data requirements include:

- The Python code should be capable of web scraping from the CVE Mitre site.
- The Python code should be capable of web scraping from a given google search.
- The Python code should be able to compare the keyword list to the descriptions.
- The HTML should be able to read the div on the scores screen to print a PDF.
- The database should hold the user's account information.
- The database should hold the user's application information.

2.1.3. User Requirements

User requirements include:

- The user should be able to register to the application.
- The user should be able to log into the application.
- The user should be able to edit their profile information.
- The user should be able to logout of the application.
- The user should be able to search for CVEs by keyword.
- The user should be able to add applications they use to the profile section.
- The user should be able to perform a quick search with their logged applications.
- The user should be able to get a report on the CVE they search for.
- The user should be able to print the report to a PDF for documentation.

2.1.4. Usability Requirements

Usability requirements include:

- The web page will load properly.
- The web page should respond to user inputs.
- The web page should be able to navigate to each HTML page cleanly.
- The database successfully connecting with the web app when running.
- The database should respond to calls from the web page about selecting data from tables.
- The database should correctly input values given to it from the web page.
- The Python code should web scrape the correct code when called.
- The Python code should listen to inputs from the user on the web page.

2.2. Design & Architecture

The main languages behind this application are HTML, CSS, Python and MySQL and I used Visual Studio Code as the IDE.

The application is broken down into its separate parts, with HTML and CSS being the UI which the user will operate from and view their results. Python and MySQL are the background engines performing the code and storing it for the user to call upon.

The Python aspect of my code is where all the functions of my code are coded and run from, the application itself runs from the Python code. I am utilising Flask, which is a package available to download for free, that allows Python to run and communicate HTML code, beautifulsoup and SERP API, web scraper APIs that allows me to get data from websites and google searches using python code. With Flask I am able to run the web scraping needed to gather the CVE's and the specific data I need to represent them. I am also able to code the session function in the python using Flask and an encryption package using Fernet. My Python code also hold my scoring code to generate a score for each of the different CVEs.

My database is made in MySQL, it holds a table for the user's accounts and the applications the users want to save on the web application. My database communicates with my Python code to store user's inputs and also to display on the HTML with data stored in the database.

I implemented a piece of JavaScript in my code, I needed this to implement a way to generate a PDF with the score page in it. This method was ideal as it took the HTML div, it makes use of the HTML2PDF package for JavaScript.

2.3. Implementation

The make-up of my code changed drastically from my mid-point submission and my final submission due to a realisation that the code I was working on had become incredibly hard and difficult to work on due to constant changes and poor maintenance on my part. So I decided to start my code over with a clear idea of what I was aiming to create and a solid understanding of how to make it with a clean slate. There is one major change that had to be made and that was to abandon the idea of using JavaScript as a major contributing factor of my code, which can be seen in how I attempted to create my basic webpage in the first submission. I replaced that with a full reliance on HTML, CSS, and Python for my project, with the option to use JavaScript for some scripting if needed and calls for it.

App.py

```
from flask import Flask, render_template, request, redirect, url_for, session
from flask_mysql import MySQL
from cryptography.fernet import Fernet
from config import SECRET_KEY
from config import app_key
import MySQLdb
import requests
from bs4 import BeautifulSoup
from serpapi import GoogleSearch

app = Flask(__name__)

#Connection for database
app.config["MYSQL_HOST"] = "localhost"
app.config["MYSQL_USER"] = "root"
app.config["MYSQL_PASSWORD"] = "mysql123456"
app.config["MYSQL_DB"] = "snooper"

db = MySQL(app)
app.secret_key = (app_key)
fernet = Fernet(SECRET_KEY)
```

Above is the start of the code for my main class that I run for the application and it is called 'app.py'. This is the main class that contains all my scripting, it is run in python and to communicate with the HTML I am using Flask, an import web framework made to work with Python. From the small picture above the main two things that can be seen is the imports and the database configuration, I will discuss the imports as they appear in other parts of the code, while the database configuration is set up here to set up the connection to the database which holds my users information.

Routes

```
# Routes to access the different pages of my application.
@app.route('/dashboard')
def dash():
    encrypted_id = session.get('user_id')
    if encrypted_id is None:
        return redirect(url_for('index'))
    return render_template('dashboard.html')

@app.route('/about')
def about():
    encrypted_id = session.get('user_id')
    if encrypted_id is None:
        return redirect(url_for('index'))
    return render_template('about.html')

@app.route('/logout')
def logout():
    session.pop('user_id', None)
    return redirect(url_for('index'))
```

Below the database configuration is the routing for the different web pages. These route functions are used in my navbar for the HTML, these act as an easy way to navigate the web application, as well as the logout button. Under the dash() and about() routes is code which reads the session to ensure that the user is logged in and it finds their 'user_id' variable, if this isn't found the code redirects the user to the 'index' function which is the sign-in function for this application. These also make use of the render_template, redirect and url_for aspect of

flask. These features allowed me as a developer to link my routes that I was coding to the HTML pages that correspond to them such as the /about route being linked to the about.html page. These made it very easy to navigate and were used all throughout the code, even when dealing with the scoring and results.

Sign-In (Index) and Registration

```
# Code for the log in and sign up with encryption and decryption on password.
@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        if 'email' in request.form and 'password' in request.form:
            email = request.form ['email']
            password = request.form ['password']
            cursor= db.connection.cursor (MySQLdb.cursors.DictCursor)
            cursor.execute ("SELECT * FROM users WHERE email=%s", (email,))
            user = cursor.fetchone ()
            if user is not None:
                encrypted_password = user['password']
                decrypted_password = fernet.decrypt(encrypted_password).decode()
                if decrypted_password == password:
                    user_id = user['my_id']
                    encrypted_id = fernet.encrypt(str(user_id).encode()).decode()
                    session['user_id'] = encrypted_id
                    return redirect (url_for('dash'))
            else:
                return redirect (url_for('index'))

    return render_template("login.html")

@app.route('/register', methods=['GET', 'POST'])
def new_user():
    if "name" in request.form and "email" in request.form and "pass" in request.form:
        name = request.form ['name']
        email = request.form ['email']
        password = request.form ['pass']
        encrypted_password = fernet.encrypt(password.encode())
        cursor = db.connection.cursor (MySQLdb.cursors.DictCursor)
        cursor.execute("INSERT INTO snooper.users (email, password, name, date_created) VALUES (%s, %s, %s, NOW())", (email, encrypted_password, name))
        db.connection.commit()
        return redirect (url_for('index'))
    return render_template("register.html")
```

The next part of the code is the aforementioned 'index' function, this is my sign-in function for my application, and I will refer to it as sign-in for this report. The sign-in function makes use of flask's request feature, and the first aspects of my database can be seen. The request feature takes from forms that have a given id such as 'email', they then connect with the database and find any users that have the same given information and the sign-in info match the user is then allowed access to the application. If there is no matching info the user is not allowed access to the application, and they are stuck on the sign-in screen. The passwords used are encrypted when the account is created, so a decryption method is used above using Fernet, I have a decryption key saved to a separate file that is called upon in the imports and above to decrypt the passwords to ensure that the users details are correct and also kept safe. The session is also

created here and can be seen taking the 'my_id' variable from the database and calling 'user_id' then it is encrypted so it can't be seen, then it is set as the session variable. The code above also shows the code for registering a new user, it uses request forms again and communicates with the database to input what is in these request forms into the database as a new user. It also encrypts the password using the fernet key. This function uses similar code to the sign-in code and however it uses different code for the database, instead of selecting the different variables, the code is inserting variables into the database.

Database

```
DROP DATABASE Snooper;
CREATE DATABASE Snooper;
USE Snooper;

CREATE TABLE users(
  my_id int auto_increment,
  email varchar(250) NOT NULL,
  password varchar(250) NOT NULL,
  name varchar(250) NOT NULL,
  company_name varchar(250),
  date_created DATETIME DEFAULT current_timestamp,
  primary key (my_id)
);

CREATE TABLE apps(
  id int auto_increment,
  user_id int,
  app_name varchar(50),
  PRIMARY KEY (id),
  FOREIGN KEY (user_id) REFERENCES users(my_id)
);
```

The code above is the query that builds and creates the tables I need for my project in MySQL. In my database I need a table to hold my users and their information, this holds the rows of my_id (users id number), email (users given email), password (users password which is encrypted), name (users given name), company_name (the company name of the user), date_created (date that their account is made). The id is given as the primary key and this links to the other table, the apps table, to designate and differentiate which input is what users. The other table needed is the apps table, this one is simple as it holds the applications that a user inputs as applications that they use in their company. It has an id row (to differentiate between the inputted values), user_id (which is the foreign key for the my_id key from the other table), and finally app_name (which is the row that contains all the applications that are inputted and sent to the database).

Profile Page functions

```
# Code below is used to get the users information in the database and display it on the page.
@app.route('/profile', methods=['GET', 'POST'])
def profile():
    encrypted_id = session.get('user_id')
    if encrypted_id is None:
        return redirect(url_for('index'))

    id = int(fernet.decrypt(encrypted_id.encode()).decode())
    cursor = db.connection.cursor(MySQLdb.cursors.DictCursor)
    cursor.execute(f"SELECT * FROM users WHERE my_id={id}")
    user1 = cursor.fetchone()
    cursor.execute(f"SELECT * FROM apps WHERE user_id={id}")
    user2 = cursor.fetchall()

    name = user1['name']
    email = user1['email']
    date = user1['date_created']
    company = user1['company_name']

    if request.method == 'POST':
        app_name = request.form['app_name']
        cursor = db.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("INSERT INTO apps (user_id, app_name) VALUES (%s, %s)", (id, app_name))
        db.connection.commit()
        ##

    return render_template('profile.html', name=name, email=email, date=date, company=company, user2=user2)

# Code below is used to change the users variables in the database.
@app.route('/edit', methods=['GET', 'POST'])
def change():
    if "name" in request.form and "email" in request.form and "company" in request.form:
        name = request.form ['name']
        email = request.form ['email']
        company = request.form ['company']
        cursor = db.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute(f"UPDATE users SET email=%s, name=%s, company_name=%s", (email, name, company))
        db.connection.commit()
        return render_template('dashboard.html')
    return render_template('edit.html')
```

The above code relates to the two functions that are found in the profile page, at the top the checks to ensure that the user is signed in are done to ensure they have a valid session open. Then a connection to the database is made where the code calls for all the users' information that is stored in the database to be taken, it is then held by the 'user1' variable and is then split into different variables depending on the rows in the database. They are then called upon and return the variables given like the name and email address provided. This code allows for the users' specific details to be given when they access their profile page. There is also a function to input data to the apps tables, users input the application they use in the form and the code connects with the database and inserts the application into the table with the user id number to ensure that it is specific to that user.

The code below that further is the function for an edit feature, this allows users to change their name, email, and company name that they have provided to the database. The code takes whatever is inputted into the forms with the specific names, it then connects with the database and updates the table with the new data.

Dashboard Search

```
# Code for the search function, makes use of an API to search for CVEs.
@app.route('/dashboard/search', methods=['GET', 'POST'])
def search_cve():
    if request.method == 'POST':
        searchword = request.form['searchkeyword']
        url = f"https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword={searchword}+2023"
        response = requests.get(url)
        if response.status_code != 200:
            print(f"Request failed with status code {response.status_code}")
            return render_template('dashboard.html', results=[])
        try:
            results = []
            soup = BeautifulSoup(response.text, 'html.parser')
            cve_table = soup.find_all('table')[2]
            for row in cve_table.find_all('tr')[1:]:
                cols = row.find_all('td')
                cve_id = cols[0].find('a').text
                description = cols[1].text.strip()
                results.append({'id': cve_id, 'summary': description})
            return render_template('dashboard.html', results=results)
        except ValueError as e:
            print(f"Failed to decode response as JSON: {response.content}")
            return render_template('dashboard.html', results=[])
    else:
        return render_template('dashboard.html')
```

The dashboard search feature is one of the main items in the project and is one of the first things seen when a user signs in. This feature in the code is laid out above. The user inputs a search word into the form, this is taken and then put into the URL for the CVE Mitre search bar, and this return a list of multiples that match the keyword in the year 2023. There is a piece of code that checks to ensure the website was able to be found and connected with, if the connection were to fail a message would be printed into the console and the user will be brought back to the dashboard with no results. If the connection does work it begins the web scraping process, it sets up a 'results' variable to hold all the data, it then calls the web scraping API and then sets up the 'cve_table' where the data will be organised. It finds the rows on the page, the first one being the CVE ID and the next being the description of the CVE ID, it puts these into variable called 'cve_id' and 'description' where there are put into the original 'results' variable to store it all until it is called by the HTML. There is an exception for a 'ValueError' in case the web scraping returns the data in a way that can't be decode by JSON. There is a very similar snippet of code that works similar but has a different start, this is my quick search feature, it utilises some of the same code when web scraping the data however it does have some differing aspects.

```

@app.route('/dashboard/quicksearch', methods=['GET', 'POST'])
def quick_search():
    encrypted_id = session.get('user_id')
    if request.method == 'POST':
        user_id = int(fernet.decrypt(encrypted_id.encode()).decode())
        cursor = db.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute(f"SELECT * FROM apps WHERE user_id={user_id}")
        quicksearch = cursor.fetchone()
        url = f"https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword={quicksearch}+2023"
        response = requests.get(url)
        if quicksearch == ():
            return render_template('dashboard.html')
        try:

```

Above is the start of the quick search feature, below the 'try' code is the same as the regular search feature. The part that sets this apart is the way it uses the database to search. From the 'apps' table, which can be populated from the profile page, the code connects to the database and selects all the 'app_names' that were inputted previously and they are put into a variable called 'quicksearch'. Once this variable is called upon in the HTML, the 'quicksearch' is added to the same URL as before and it collects the data then. This feature was added to allow for the user to input their personal applications that they use and then be able to quickly search for them with ease without needing to type in individually the applications and software.

Web Scrape Scores

```

#Scrape Google for the total search results of CVE's
params = {
    "engine": "google",
    "q": cve_id,
    "api_key": 'aa4e0796efda079482a2f396a4c38455611928b4b843bf2c43a94685f564fcd5'
}
try:
    search = GoogleSearch(params)
    result = search.get_dict()
    if "organic_results" in result and len(result["organic_results"]) > 0:
        search_stats_str = result["search_information"]["total_results"]
        search_stats_int = int(search_stats_str)
        google = f"{search_stats_int}"
    else:
        google = "N/A"
except Exception as e:
    print(f"Error retrieving Google search results for {cve_id}: {str(e)}")
    google = "Error"

#Scrape the NIST site to see if CVE's are being exploited
response = requests.get(nist_url)
if response.status_code != 200:
    print(f"Request failed with status code {response.status_code}")
    exploit = "Error"
else:
    soup = BeautifulSoup(response.text, 'html.parser')
    exploit_result = soup.find(id="vulnCisaExploit")
    if exploit_result is not None:
        exploit = "Being Exploited"
    else:
        exploit = "Not Exploited"

```

The code above is a snippet of code used to scrape websites for their scores much of the code is similar, so I chose to show two snippets that are a good representation of the work that is being done. The first bit of code is the snippet that scrapes a google search, it uses the SERP API to perform this as this API is made for scraping google searches and ensures that it is done in a way where the scraper is not going against any of googles terms. Firstly the parameters are set, the search engine being used is google, the query being searched is the 'cve_id' variable, which is taken from the search feature previously mentioned, finally the last parameter is the API key which allows the API to work and scrape the google search. The parameters are inputted into the search and the results are taken and stored in the 'result' variable. Then the code checks for 'organic_results' and that some were actually found, these are defined in the SERP API as results in google that are unique for every search meaning if you were to search for something multiple times, each time the HTML divs have different naming conventions. If the search returned some results, then the code finds the divs that relate to the 'search_information' and further the 'total_results'. These naming conventions are defined in SERP API as well as parts of the google search web scraper and help to easily find certain elements on the page. The results will be returned as a string, however for my project I need it to be an integer, so it is changed to the right variable. Finally it is held in the 'google' variable till it is called to be produced in the HTML.

The next snippet of code I used another few times however due to the similarities I have decided to just show this snippet, however the rest can be seen in my code with the only real differences being that the div id's that they are searching for are different. For this code in question, firstly the code calls to open a connection to the 'nist_url' which is a variable predefined with the URL for the NIST website and the needed CVE ID at the end. There is a check to ensure that it connects properly and once that is passed the web scraper beautifulsoup is used. The web scraper is called upon and it searches for the given div id, which in this case is 'vulnCisaExploit'. If the web scraper finds the div in question it returns that the CVE is being exploited. This is due to the fact that the CVE page has links to the CISA database which keeps tabs on what CVE's are being exploited at this moment in time. If the div is not found, then it is returned that the CVE is not being exploited at this moment.

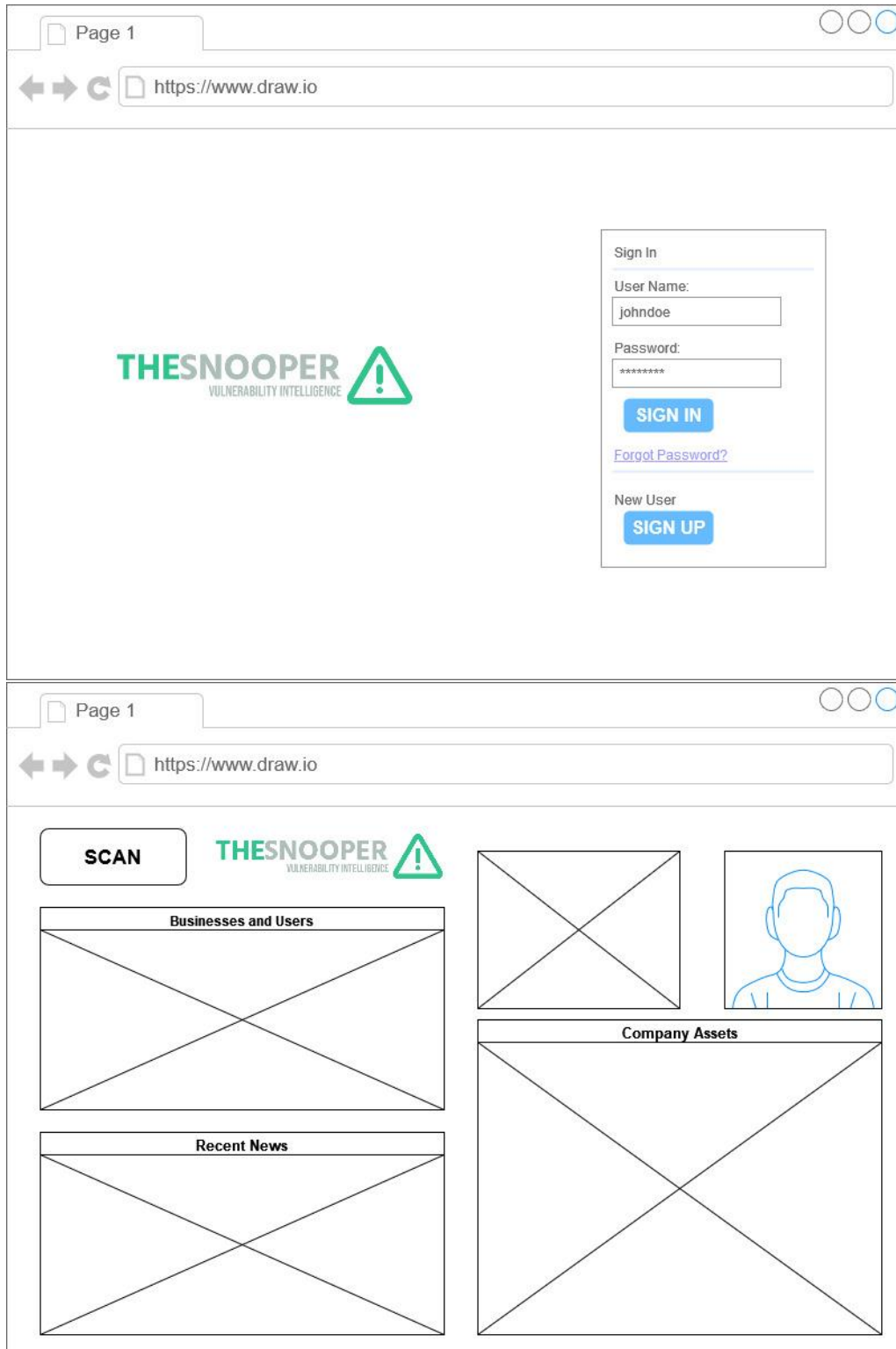
Generate PDF

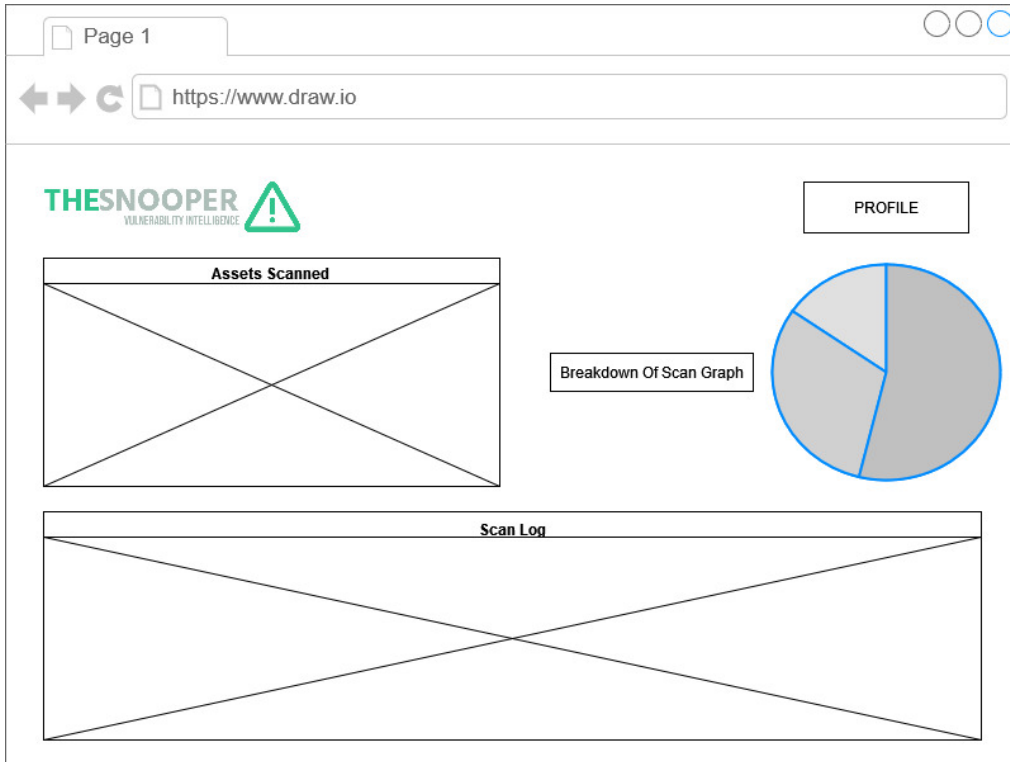
```
<script src="https://cdnjs.cloudflare.com/ajax/libs/html2pdf.js/0.9.2/html2pdf.bundle.min.js"></script>
<script>
  function generatePDF() {
    var element = document.getElementById('scores');
    var now = new Date().toLocaleString().replace(',','').replace(/:.. /, ' ');
    var filename = 'result ' + now + '.pdf';
    var opt = {
      margin:      0,
      filename:    filename,
      image:       { type: 'jpeg', quality: 0.98 },
      html2canvas: { scale: 2 },
      jsPDF:       { unit: 'in', format: 'letter', orientation: 'portrait' }
    };
    html2pdf().set(opt).from(element).save();
  }
</script>
```

The code above is the only piece of JavaScript in the project. Due to many difficulties getting Python PDF generation APIs working, I resorted to using a JavaScript one that I had been testing and using in the first iteration of my code. The code uses a bundle called HTML2PDF which using some code allows PDFs to be generated with pieces of HTML being printed in them. What I have done in this instance is grabbed the HTML element 'score' by designating it with that ID name. I

then set up a variable which gets the time and date to input it into the name of the PDF so users can differentiate between PDFs. I then set some options for how the PDF will be made up, such as making the orientation portrait and the units as inches. I then made it to the PDF when called upon set the options and grabbed the element 'scores' when it was called on.

2.4. Graphical User Interface (GUI)

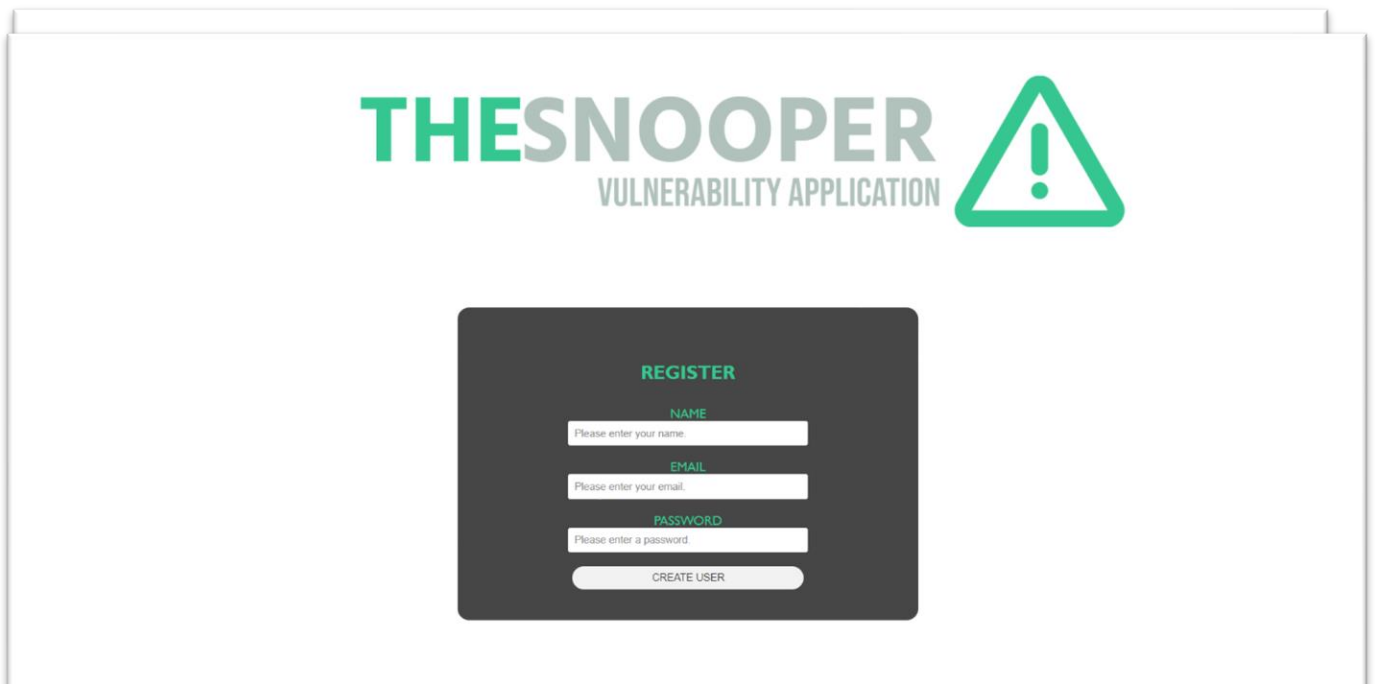




Above is

provisional designs and wireframes of the UI: 1st Log in Screen, 2nd User Profile, 3rd Dashboard. Below are the screenshots of how the UI turned out in the end.

The above screenshot is of the first screen you see when you open the web application. You are met with this screen where you can input your login details and proceed, or you can press the register button to create an account.



Next is the registration screen, this is similar to the log in screen. You input details to create an account and then once you confirm by pressing the 'create user' button, you are brought back to the log in screen.

The screenshot shows the Snoopier dashboard with a navigation bar (Home, Profile, About, Logout) and a search interface. A table displays search results for CVEs, and a sidebar titled 'The Hacker News' contains several news items.

CVE ID	Description	Analyse
CVE-2023-30610	aws-sigv4 is a rust library for low level request signing in the aws cloud platform. The 'aws_sigv4::SigningParams' struct had a derived 'Debug' implementation. When debug-formatted, it would include a user's AWS access key, AWS secret key, and security token in plaintext. When TRACE-level logging is enabled for an SDK, 'SigningParams' is printed, thereby revealing those credentials to anyone with access to logs. All users of the AWS SDK for Rust who enabled TRACE-level logging, either globally (e.g. 'RUST_LOG=trace'), or for the 'aws-sigv4' crate specifically are affected. This issue has been addressed in a set of new releases. Users are advised to upgrade. Users unable to upgrade should disable TRACE-level logging for AWS Rust SDK crates.	Score
CVE-2023-29010	Budibase is a low code platform for creating internal tools, workflows, and admin panels. Versions prior to 2.4.3 (07 March 2023) are vulnerable to Server-Side Request Forgery. This can lead to an attacker gaining access to a Budibase AWS secret key. Users of Budibase cloud need to take no action. Self-host users who run Budibase on the public internet and are using a cloud provider that allows HTTP access to metadata information should ensure that when they deploy Budibase live, their internal metadata	Score

The Hacker News

- New Flaw in WordPress Plugin Used by Over a Million Sites**
A security vulnerability has been disclosed in the popular WordPress plugin Essential Addons for Elementor...
May 12, 2023 6:43 AM
- New APT Group Red Stinger Targets Military and Critical**
A previously undetected advanced persistent threat (APT) actor dubbed Red Stinger has been linked to...
May 11, 2023 3:45 PM
- How Attack Surface Management Supports Continuous Threat Exposure Management**
According to Forrester, External Attack Surface Management (EASM) emerged as a market category in 20...
May 11, 2023 3:05 PM
- Spanish Police Takes Down Massive Cybercrime Ring, 40 A...**
The National Police of Spain said it arrested 40 individuals for their alleged involvement in an org...
May 11, 2023 3:00 PM

The next screenshot is of the dashboard, a search has been performed to show what a user will see when that happens. This is the screen you are directed to when you first log in, this is where the majority of the work will be done by the user. The nav bar at the top allow the user to navigate to the other web pages and to log out too. There is an RSS feed of a prominent Cybersecurity new source, "The Hacker News", to the right of the page, this is also found on the profile and about page. The search form, search button and quick search button can be seen at the top above the table. The table can be seen showing the CVE ID, the description and give the option to score the CVE's separately with the button to the right of the table.

The screenshot shows the Snoopier profile page. The navigation bar includes Home, Profile (active), About, and Logout. The profile information is displayed in a table, and there is a section for 'Applications Used' with a search input and a submit button. A sidebar titled 'The Hacker News' is visible on the right.

Profile:

Name:	test
Email:	test@work.ie
Company:	Test Company
Date Joined:	2023-05-11 16:37:37
Reports:	N/A

Applications Used:

Please enter applications you use

Microsoft Office

gmail

AWS

The Hacker News

- Solving Your Teams Secure Collaboration Challenges**
In today's interconnected world, where organisations regularly exchange sensitive information with c...
May 12, 2023 12:33 PM
- BROODy Ransomware Gang Strikes Education Sector with Cr**
U.S. cybersecurity and intelligence agencies have warned of attacks carried out by a threat actor kn...
May 12, 2023 8:59 AM
- New Flaw in WordPress Plugin Used by Over a Million Sites**
A security vulnerability has been disclosed in the popular WordPress plugin Essential Addons for Ele...
May 12, 2023 5:43 AM
- New APT Group Red Stinger Targets Military and Critical**
A previously undetected advanced persistent threat (APT) actor dubbed Red Stinger has been linked to...
May 11, 2023 3:45 PM

Powered by feedwind

The next screenshot above is of the Profile screen and here users can see their profile details such as their name, email address, company name, date the account was created and their previous reports. You can also see at the bottom of the page is the applications used table, here a user inputs an application they use into the form and then it is sent to the database to be stored and shows up below in the table.

Home Profile **About** Logout

THE SNOOPER
VULNERABILITY APPLICATION

About Project

This product uses the MitreCVEAPI but is not endorsed or certified by Mitre.

This web application is made with small/medium size businesses in mind. It is made to help organisation to discover vulnerabilities that may affect their systems.

This is done by searching for CVE's on the CVE Mitre website and collecting the data related to them there. Then when you found a CVE you want more information on, you press the 'score' button beside it to get more details. Here you can see the scoring metrics such as CVSS 3 and 2 scores if they have been designated scores.

Here you can also see two more metrics are in use, 'Exploited' and 'Google Search', these were chosen for similar reasons. If a CVE is being exploited it means it is going to be more dangerous to an organisation compared to one that is not being exploited. For the Google Search, this metric is taken due to the public interest that some CVE's gain, the more people discussing it and posting articles or mention it is news sites means there is something to talk about meaning it could be dangerous to an organisation. All in all these metrics combine into what I call the Snooper Score. It is a scoring system that I have created, it places more emphasis on the CVE's being exploited and the CVSS 3 score, while Google searches and CVSS 2 are additional scores that give more information to the user.

How the scoring works:

CVSS 3: 'Low' = 2 'Medium' = 4 'High' = 6 'Critical' = 8

CVSS 2: 'Low' = 1 'Medium' = 2 'High' = 3

Exploited: 'Being Exploited' = 5 'Not Exploited' = 0

Google Search: '1-999' = 1 '1000-9999' = 2 '10000-99999' = 3 '100000+' = 4

They are then added together to give a snooper score.

The Hacker News

- Solving Your Teams Secure Collaboration Challenges**
In today's interconnected world, where organisations regularly exchange sensitive information with c...
May 12, 2023 12:33 PM
- Bi00dy Ransomware Gang Strikes Education Sector with Cr...**
U.S. cybersecurity and intelligence agencies have warned of attacks carried out by a threat actor kn...
May 12, 2023 8:59 AM
- New Flaw in WordPress Plugin Used by Over a Million Sit...**
A security vulnerability has been disclosed in the popular WordPress plugin Essential Addons for Ele...
May 12, 2023 6:43 AM
- New APT Group Red Stinger Targets Military and Critical...**
A previously undetected advanced persistent threat (APT) actor dubbed Red Stinger has been linked to...
May 11, 2023 3:45 PM

Powered by feedwind

The About screen is next, here I give some information on the application, such as why I made the application, some reasons as to why I used some of the scoring metrics and how the scoring works.

CVE-2023-25956

Description

Generation of Error Message Containing Sensitive Information vulnerability in the Apache Airflow AWS Provider. This issue affects Apache Airflow AWS Provider versions before 7.2.1.

Link: [CVE LINK](#)

Keywords:

CVSS 3	CVSS 2	Exploited	Google	Snooper Score
7.5 HIGH	N/A	Not Exploited	2490	8

Result Score is below 10, use caution when dealing with this CVE as it could pose a problem in the future. To learn more about the scoring system please refer to the about page.

THE SNOOPER
VULNERABILITY APPLICATION

Generate PDF

The final screenshot is the screenshot of the Score screen. This is what is shown after the user presses the 'score' button on the dashboard screen. The CVE that was chosen is shown at the top of the screen with its ID and description being given, a link to the NIST website is given for users to access the CVE from there. A keyword option is given, where if a keyword is found in the description, it is shown again here to highlight to the user of the main issue with this CVE. Next is the scoring system, here is where the web scrapers data is used, it takes the data for each of the variables, CVSS 3, CVSS 2, Exploitation and Google searches and displays them for the user. Then my personal Snooper Score is involved. Each of the scoring metrics are given a score depending on what they return. CVSS 3 has four main outputs 'X.X LOW', 'X.X MEDIUM', 'X.X HIGH', 'X.X CRITICAL', the 'X's account for the number they return with. For the Snooper Score, these are given variables of LOW=2, MEDIUM=4, HIGH=6 and CRITICAL=8. CVSS 2 is similar however it only uses three outputs, 'X.X LOW', 'X.X MEDIUM' and 'X.X HIGH'. These are given a score of LOW=1, MEDIUM=2 and HIGH=3, these scores are lower than the scores given to CVSS 3 due to CVSS 2 not being utilised as much anymore, so they are not given as much weight for the newer CVE's. Exploited is treated as a Boolean variable, either the CVE is being exploited or it is not so for that reason the scores for that are 'Being Exploited'=5 and 'Not Exploited'=0. This is given a high score due to my opinion of CVE's being exploited no matter the CVSS score is important as it means attackers are using it to hack organizations. Finally the google search results, I decided to use this metric due to my understanding that the more people and media sources are discussing a given CVE and topic, it means that there is something that needs to be monitored for this item. The scoring for this goes as follows, less than 1000 searches =1, between 1000 and 10000 searches =2, between 10000 and 100000 searches =3, and anything above 100000 searches =4. These given scores are then added together to give the total Snooper Score. Depending on the score a little message is given below to give some users some insight into how serious to take it.

2.5. Testing

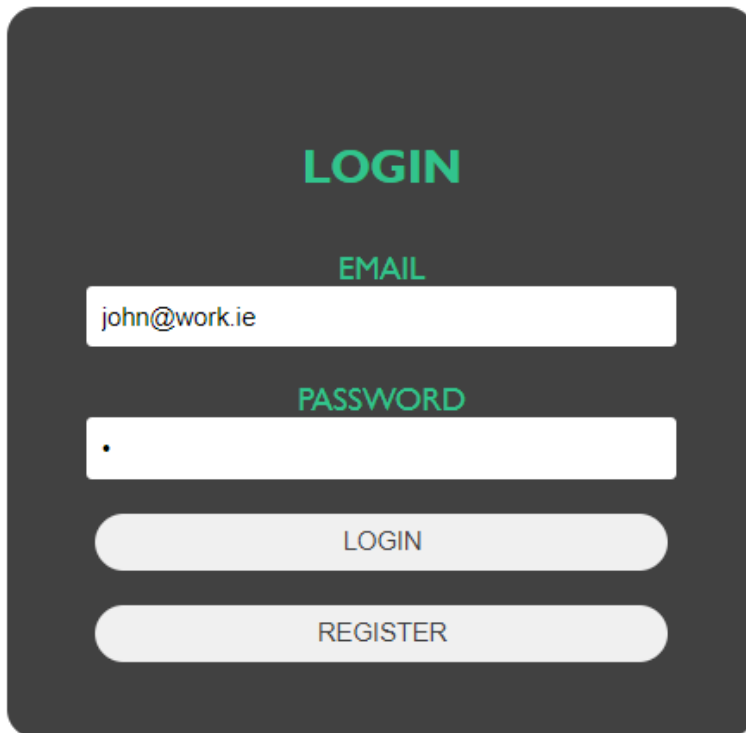
At the end of the project I will be creating test cases, some with data and log ins that are correct, I will also throw in some test cases where data is in the incorrect fields or looking in the wrong table of the database.

Log In Testing

For the first test, I want to figure out if my system log in works as intended. This test case will be divided into two parts the first is an incorrect log in and the second is a correct log in.

1. Incorrect Log In

For my incorrect log in, I will be inputting an incorrect email and password as well to check if the system picks up on the incorrect inputs. The expected outcome of this is for the application to reject the log in details and reload the log in screen with a message being printed in the application console.

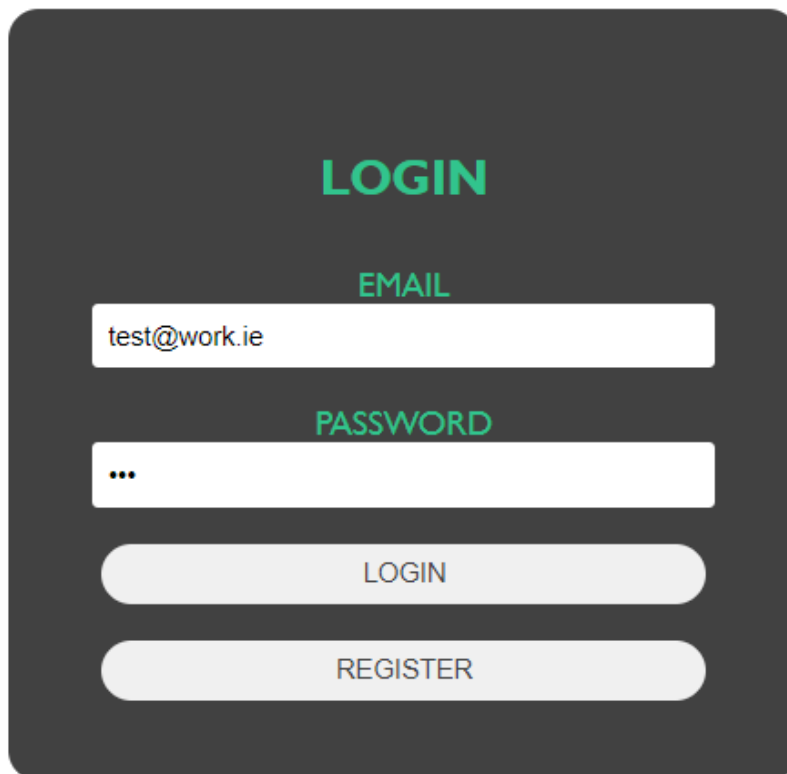


Above is a screenshot of my log in screen with the email being set to 'john@work.ie' and the password is '1'. When the user presses the log in screen a message is printed into the console of the application on whether the input was successful or not. Below is a screenshot of the console with the message of 'Log in unsuccessful', the code was able to correctly see the code as wrong and didn't allow access to the application.

```
127.0.0.1 - - [13/May/2023 10:59:05] "GET /static/css/login.css HTTP/1.1" 304 -
127.0.0.1 - - [13/May/2023 10:59:05] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [13/May/2023 10:59:11] "GET /static/css/login.css HTTP/1.1" 304 -
Log in unsuccessful
```

2. Correct Log In

For my correct log in I will give a known correct email address and the password that was created with it. This is to show what happens to the application when it is given a correct log in detail. The expected outcome is for the application to accept the log in details and to send the user to the dashboard page with a message confirming the action in the application console.



The above screenshot is of the log in screen with the email input being 'test@work.ie' and the password being '123'. The user presses the log in screen and when the user does this they are brought to the dashboard of the main application. The code also prints a message saying 'Log in successful' in the console. This can be seen below.

```
127.0.0.1 - - [13/May/2023 11:23:19] "GET /static/logos/logo-nb.png HTTP/1.1" 304 -
127.0.0.1 - - [13/May/2023 11:23:20] "GET / HTTP/1.1" 200 -
Log in successful
127.0.0.1 - - [13/May/2023 11:25:09] "POST / HTTP/1.1" 302 -
```

In regard to testing the log in capability of the application, the system prints the error when a user incorrectly Logs in, while prints a success message in the console. The test has proved successful in showing this feature.

Search Feature Test

This test is to check the integrity of the main feature of my project and that is the CVE Keyword search feature. This finds the CVEs for a user and helps them to find ones they will need to combat. This section will be split into two parts, first will be dealing with the CVE Keyword Search and the second will deal with the Quick Search feature.

1. CVE Keyword Search

The keyword search is one of the main features of this project and it needs to be running correctly for the project to work. The first aspect I will test is ensuring that a keyword must be inputted into the search bar for the search function to run. This can be seen

below, if a user were to press the 'Run' button with an empty bar, a message should be displayed in the console and the user should be left with an empty dashboard table.



This can be seen here, the message is printed in the code, this means the search function did not run and the user was left with no results. This helps to prevent unnecessary strain on the system having to search for all CVEs.

```
127.0.0.1 - - [13/May/2023 11:41:01] "GET /static/images/logo-hb.png HTTP/1.1" 200 -
127.0.0.1 - - [13/May/2023 11:41:01] "GET /dashboard HTTP/1.1" 200 -
No keyword entered, can't search.
127.0.0.1 - - [13/May/2023 11:41:10] "POST /dashboard/search HTTP/1.1" 200 -
```

On the other hand when a user does input a keyword in the search bar, the search function should run with a message being displayed in the console stating so, the user will be then showed the results of their search.

To test this, I am inputting a keyword search for 'AWS' and I am expecting the search to run, a message to be printed in the console and for the results to be printed onto the dashboard table.



Above can be seen the 'AWS' input into the search bar.

```
CVE search complete.
127.0.0.1 - - [13/May/2023 12:10:08] "POST /dashboard/search HTTP/1.1" 200 -
```

After the button is pressed, the message is displayed to say the CVE search has been completed and the results are displayed on the dashboard page in the table, an example of this can be seen below.

CVE ID	Description	Analyse
CVE-2023-30610	aws-sigv4 is a rust library for low level request signing in the aws cloud platform. The 'aws_sigv4:SigningParams' struct had a derived 'Debug' implementation. When debug-formatted, it would include a user's AWS access key, AWS secret key, and security token in plaintext. When TRACE-level logging is enabled for an SDK, 'SigningParams' is printed, thereby revealing those credentials to anyone with access to logs. All users of the AWS SDK for Rust who enabled TRACE-level logging, either globally (e.g. 'RUST_LOG=trace'), or for the 'aws-sigv4' crate specifically are affected. This issue has been addressed in a set of new releases. Users are advised to upgrade. Users unable to upgrade should disable TRACE-level logging for AWS Rust SDK crates.	Score
CVE-2023-29010	Budibase is a low code platform for creating internal tools, workflows, and admin panels. Versions prior to 2.4.3 (07 March 2023) are vulnerable to Server-Side Request Forgery. This can lead to an attacker gaining access to a Budibase AWS secret key. Users of Budibase cloud need to take no action. Self-host users who run Budibase on the public internet and are using a cloud provider that allows HTTP access to metadata information should ensure that when they deploy Budibase live, their internal metadata endpoint is not exposed.	Score

2. Quick Search

For my quick search feature, I am testing it by pressing the 'Quick Search' button. I am expecting, the code to run and send the apps inputted by the user in the database to be searched for, then a line is printed into the console to confirm it works.

RUN

QUICK SEARCH

The user presses the button, with 'Microsoft Office' being the app inputted into the database. The code runs the quick search function and outputs the message to the console as seen below.

```
Quick search complete.  
127.0.0.1 - - [13/May/2023 12:24:36] "POST /dashboard/quicksearch HTTP/1.1" 200 -
```

The results from the search are then displayed in the results table on the dashboard.

CVE ID	Description	Analyse
CVE-2017-8744	A remote code execution vulnerability exists in Excel Services, Microsoft Excel 2007 Service Pack 3, Microsoft Excel 2010 Service Pack 2, Microsoft Excel 2013 Service Pack 1, Microsoft Excel 2013 RT Service Pack 1, and Microsoft Excel 2016 when they fail to properly handle objects in memory, aka "Microsoft Office Memory Corruption Vulnerability". This CVE ID is unique from CVE-2017-8630, CVE-2017-8632, and CVE-2017-8731.	Score
CVE-2017-8742	A remote code execution vulnerability exists in Microsoft PowerPoint 2007 Service Pack 3, Microsoft PowerPoint 2010 Service Pack 2, Microsoft PowerPoint 2013 Service Pack 1, Microsoft PowerPoint 2013 RT Service Pack 1, Microsoft PowerPoint 2016, Microsoft PowerPoint Viewer 2007, Microsoft SharePoint Server 2013 Service Pack 1, Microsoft SharePoint Enterprise Server 2016, Microsoft Office Web Apps 2010 Service Pack 2, and Microsoft Office Compatibility Pack Service Pack 3 when they fail to	Score

Profile Page Test

The profile page of the web application is a hub for the user's information, here they can view the information they have given, edit it and to add applications they use to the database. I want to test the application input feature and the edit profile feature.

1. User Applications Input Test

The user has the option to input applications and software they personally use into the projects database, this allows them to make use of the quick search function.

To test this, the user inputs a software or application they make use of into the input bar on the profile screen. I expect the code to run and take the input given, put it into the database table and then return a message to the console confirming the action.

Applications Used:

azure

Submit

For this test the user inputted 'Gmail', 'AWS' and 'Azure' into the input box. The code sent these to the database and the console message of 'Applications successfully sent to the database.' was printed there.

```
Applications successfully sent to database.  
127.0.0.1 - - [13/May/2023 13:27:02] "POST /profile HTTP/1.1" 200 -
```

The results of the input can be seen in the table beneath the input box, and this is updated as new items are sent to the database.

Applications Used:

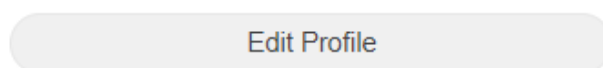
Microsoft Office

AWS

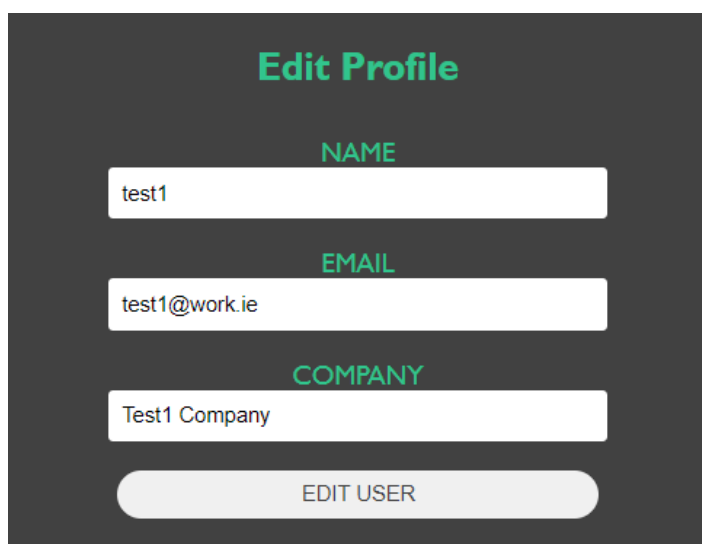
2. Edit Profile

The edit profile feature of the project is one of the minor features however it does give users more interaction with the application and allows them to somewhat personalise their accounts with things that are unique to them such as their company name.

To test this feature I will be pressing the edit profile button and changing user information when asked to, the button is seen below. Once pressed the user is brought to a page to input the new information.



The information input page takes a new name, email, and company name from the user, once entered and the button is pressed the user is then sent to dashboard. The input box is seen below in this test the name has been set to test1, email is set to 'test1@work.ie' and the company is set to 'Test1 Company'.

A dark grey form titled "Edit Profile" in green. It contains three input fields: "NAME" with "test1", "EMAIL" with "test1@work.ie", and "COMPANY" with "Test1 Company". At the bottom is a light grey rounded button labeled "EDIT USER".

Edit Profile

NAME
test1

EMAIL
test1@work.ie

COMPANY
Test1 Company

EDIT USER

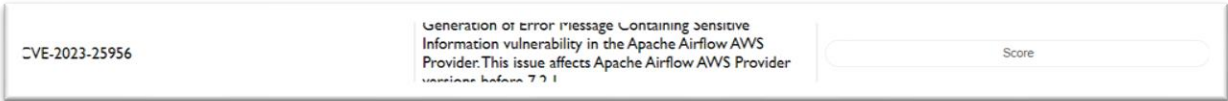
The output message in the console can be seen below and the updated information table is seen below as well.

```
127.0.0.1 - - [13/May/2023 13:50:21] "GET /edit HTTP/1.1" 200 -
User account information has been updated.
127.0.0.1 - - [13/May/2023 13:50:50] "POST /edit HTTP/1.1" 200 -
```

Name:	testl
Email:	testl@work.ie
Company:	Testl Company
Date Joined:	2023-05-11 16:37:37
Reports:	N/A

Scoring Test

The scoring aspect of the application comes after a search has been made. It involves gathering information on a given CVE and showing it to the user in a readable format. I am going to test this by running the script and checking for the correct message in the console.



The first step of this is to find a CVE to get a score of then press the score button on the left, this can be seen in the screenshot above.

In the screenshot below, the console message can be seen. It says that all the search variables have been found which include the CVSS3, CVSS2 scores, whether its being exploited and how many google search results there are.

```
CVSS3 score found.
CVSS2 score found.
CVE not being exploited.
Google search results found.
127.0.0.1 - - [13/May/2023 14:10:32] "POST /score HTTP/1.1" 200 -
```

You can see the results given in the screenshot below, this shows the score feature works and that it loads up correctly.

CVSS 3	CVSS 2	Exploited	Google	Snooper Score
7.5 HIGH	N/A	Not Exploited	2200	8

2.6. Evaluation

The web application was evaluated by running a number of the tests mentioned above. Due to the nature of the application the way to test it revolves around testing it with different variable such as incorrect information and successful information. The repeated testing of this application is the best way to check for inconsistencies I have found. An ideal way to test and evaluate the application would have been to deploy the application and test it from a multitude of systems and devices however this was not possible due to how the project developed and the time constraints of the project. With the testing I was able to do I discovered that the core features of the project worked, and I was happy with the output I was getting from them. The only test that returned some problems was the quick search test, Tests 1,2 and 5 were done with only one variable in the database, however Tests 3 and 4 had multiple variables in the database yet the results only showed one of the variables, work needs to be done to check this issue and fix it.

TEST	TEST 1	TEST 2	TEST3	TEST 4	TEST 5	OVERALL
CORECT LOG IN	Success	Success	Success	Success	Success	Every test was successful
INCORRECT LOG IN	Success	Success	Success	Success	Success	Every test was successful
CVE SEARCH	Success	Success	Success	Success	Success	Every test was successful
QUICK SEARCH	Success	Success	Fail	Fail	Success	Mixed results need to assess
EDIT PROFILE	Success	Success	Success	Success	Success	Every test was successful
APP INPUT	Success	Success	Success	Success	Success	Every test was successful
SCORING	Success	Success	Success	Success	Success	Every test was successful

3.0 Conclusions

Originally the project was started and worked on with the intention of making use of JavaScript to build the website in conjunction with the HTML and CSS, however when attempting to implement the python as the backend, the connection and use of major functions ceased. I had to make a difficult decision to not use JavaScript as my knowledge of that and python were not extensive enough to properly carry out what I expected to do and achieve, this entailed my project being restarted as no progress was being made in regard to any aspect of the project. With the project being restarted I based my project fully with python and HTML and within a few days progress to the point I was at previously was achieved. This was a major set-back however it did allow me to advance and progress in my project.

Advantages

The project advantages are numerous especially for my targeted users. With the applications search feature and scoring system it allows for users to find CVEs that relate to applications and software that they use in their company. This service provides them with intelligence and access to databases full of CVEs and vulnerabilities that they will be given knowledge of. The application provides a scoring system that will help give users some information and more knowledge of the CVEs and further the vulnerabilities linked to them. The scoring provides some intelligence, and the users are given more information allowing them to make more informed decisions on what needs to be done for their organisation. This product is aimed at helping small and medium-sized businesses to decided and prioritise what needs to be done to combat vulnerabilities their organization may face and what they should do in their next patching session.

Disadvantages

The disadvantages of this product include the project being built for PC device and not with mobile device in mind, meaning that it can't be used on other devices. The purpose of this project is quite specific meaning that it has a smaller user base that they can access. The product is not a very commercially viable product as it does not have much appeal unless users are looking for some aspect of vulnerability intelligence. The project is not deployed online also which limits the accessibility of the application to anyone who downloads and runs the application on their own personal device, this can be mitigated by deploying the product, however at this time the product isn't deployed.

4.0 Further Development or Research

With additional time and resources I would like to increase the searching capacity of the application to other CVE databases such as Vuln DB and increase the capacity of searching on NIST, I would also like to fix the issues that have arose in relation to the quick search feature and make it work as intended. This would mean the searches for CVEs would be more in-depth and allow for more precise searching. I would also increase the number of metrics to which the CVE's are scored by. Making use of Google Searches and whether the CVEs are being exploited is a necessary thing to account for so I would like to add more items to supplement it. For a small but key aspect I would like to develop is the PDF report feature, I would like to ideally be able to add them to the database for future reference and allow users to download them when needed, this allows for a system of tracking the CVEs users have previously made a report on and a simple way to find those CVEs again. I would also like to include so visual data for the user as this would give them a visual understanding of how items are scored and how they may rank on a Snooper Scale. If it were to progress enough, I would like to include instructions from the vendors about the certain CVEs so users can learn about and fix their vulnerabilities all from the service I would provide.

5.0 References

CVE, M. (2020) *MITRE CVE documentation, Mitre CVE documentation - Mitre CVE 1.0.1 documentation*. Available at: <https://mitrecve.readthedocs.io/en/latest/index.html>.

gov, N. (n.d.). *Vulnerability APIs*. [online] nvd.nist.gov. Available at: <https://nvd.nist.gov/developers/vulnerabilities#>.

Cybersecurity and Infrastructure Security Agency (n.d.). *Known Exploited Vulnerabilities Catalog | CISA*. [online] www.cisa.gov. Available at: <https://www.cisa.gov/known-exploited-vulnerabilities-catalog>.

Koopmans, E. (n.d.). *html2pdf.js*. [online] html2pdf.js. Available at: <https://ekoopmans.github.io/html2pdf.js/>.

Ronacher, A. (n.d.). *Welcome to Flask — Flask Documentation (2.3.x)*. [online] flask.palletsprojects.com. Available at: <https://flask.palletsprojects.com/en/2.3.x/>.

CVE, M. (2009). *Number Of Security Vulnerabilities By CVSS Scores*. [online] [Cvedetails.com](http://cvedetails.com). Available at: <https://www.cvedetails.com/cvss-score-distribution.php>.

6.0 Appendices

6.1. Project Proposal

I made an error during my project proposal calling it a scanner when it is not a scanner at all it is a vulnerability intelligence application, I only realised my error after being able to discuss it with my supervisor.



National College of Ireland

Project Proposal

Vulnerability Scanner for Small/Medium size Businesses

31/10/2022

Bachelor of Science in Computing (BSHC)

Cyber Security

2022/2023

Nathan Savage

18395306

x18395306@student.ncirl.ie

1.0 Objectives

The main objective I set out for this project is to help small/medium size businesses to be able to perform and avail of the service that a vulnerability manager can provide. These can include identifying threats and understanding them and how to combat them. The program I want to create will have the objective to identify vulnerabilities based on an inventory of an asset list, give concise information and intelligence on those vulnerabilities, compile internal and external factors of the vulnerability, create, and provide a report that can be used by teams to decide on what to patch to combat the vulnerability.

The small/medium size business will need the information from the report to decide how to prioritise what is patched and when. Patching can be an arduous task that could need a lot of work so the business will need all the information it can get to decide what is best for the company.

2.0 Background

I chose to undertake this project as this is a relatively new area of Cyber Security to myself, and I found it very interesting with a small bit of research and I want to continue researching and learning more about this topic.

While on my internship, every month the department would perform patching for the company. My team handled the batch updates and backups which meant these had to be paused for patching. When this was happening, I realised how this leaves vulnerabilities for the company, as if the patching went wrong the batch backups would mean some work is lost and my teams work cannot be done. This got me thinking of the other vulnerabilities to patching in the business considering the size of the company, Irish Life, and the impact it might have on their parent company, The Great West Life Co, which has offices across North America and Europe. And further the other vulnerabilities that could arise in general due to the size of the organisation. I then went on to think how a smaller business would deal with vulnerabilities as my experience shows a huge number of people being involved with many differing jobs and experience needed to make sure everything ran smoothly.

I will achieve the objectives by building a vulnerability scanner for small/medium size organisations, I will improve my understanding of the subject by simply working on the application and teaching myself about this subject to a greater extent. The best way to document this objective is done in the monthly journals and in the code itself by implementing what I learn and what I find useful.

3.0 State of the Art

The field for this technology is growing in the cyber security sector itself. There are many applications developed to try and fill the market. These include applications like Qualys and Tenable.

Qualys is a software application that is centred around network security and vulnerability management. It was founded in 1999 and focuses on SaaS (Software as a Service) and is based out of California. Some of the features include app scanning and security, network device mapping and detection and vulnerability prioritisation schedule and remediation, and more but these are some of the major ones.

Tenable is like Qualys in its function and features, but it was set up in 2002 with its headquarters in Columbia, Maryland, with an international headquarters in Dublin. Tenable offer many applications based on Cyber Security as a whole, with Tenable.io and Tenable.sc being two that focus more on vulnerability management. Tenable.io focuses mainly on web applications and scanning them and viewing the cloud infrastructure for vulnerabilities. While Tenable.sc is their main security centre that focuses on all aspects of security.

There has been a previous project done on vulnerability applications but that was done the python. The way I would like to do my application is to do my project in JavaScript and improve my knowledge of this language further.

4.0 Technical Approach

My Technical approach is going to be broken up into a multitude of steps depending on the part of the project I am working on. I plan to create a web application for this project so this in itself provides an insight into the steps that will be taken. The way I identified my requirements for my project, I broke down the application into its different parts and gave each of the major parts the label of milestone, while the smaller steps that made these, I labelled them the tasks, for example creating the application is the milestone, while getting the code to scan web pages is a task in the milestone.

I will create a web page using HTML and CSS as the directory for the application to run and for the users to use as a hub to run and view the results of the application. My first milestone will be the completion of this webpage, while the tasks include designing a secure web page, designing the web page and to optimise it for different browsers and devices.

The next milestone that I can foresee is the creation of the application itself and getting the code to run. The tasks in this would be to get a web scanner working, the website to be linked properly to the program and the program to be able to access the database that is connected to the program and is able to communicate properly and securely with each other.

The next milestone is the correct set up of a database that the program can use to store data collected and also a database that holds data so the program can compare that data

with data that it scans and collects. With the tasks being setting up the databases, giving them the correct privileges and setting up the connection properly.

5.0 Technical Details

For a majority of the work being done on this project I have not decided fully on what library and languages I will be using on my project, and I will be talking with my supervisor on what I could do. I have thought of some languages I could use for the main backbone of my code for my web app and the web page specifically, also for the database.

For my web app, I want my project to revolve around JavaScript due to the fact I have prior knowledge in the language and also, I want to further get more experience in the code. I am also open to using Python, I have worked on it before and wish to get further experienced in the language, I am still considering if I want to revolve the project around it or just use it for added scripts.

For the web page, I want to focus on HTML and CSS. I have worked on this in my first year of college and I want to relearn the language and increase my knowledge of the code and language to a further level. Finally for my database, I want to use MySQL as I feel it has a good way to interact with the code I want to make, and I have prior experience in the language.

6.0 Special Resources Required

No special resources are needed for this project.

7.0 Project Plan

The project plan is split into a timeline of three major milestones, with varying amounts of tasks that go into each of them. The main milestones being the web page, the web application, and the databases.

The first milestone being the web page is the first task I want to start with, the creation and design of the page will be my first task and I will make use of HTML and CSS to make the base and User Interface for my project. This step is the first one of my project, and I want to start on this at the start of November, I plan to have the base of the web page done with its core features completed before the end of December, I will continue to work and tweak the minor designs of the page up until the end of the project in May. More tasks that go into this is the securing of the website and ensuring that it is a safe and viable website to access without any concern for cyber-attacks or security concerns for a user. Another task that goes into this milestone is to successfully link the web application to the web page, this is a big milestone too as it connects two integral parts of the project together. Finally, the last

task for this milestone is successfully hosting the web page on the web so it can be tested to see if it works on a live setting.

The next major milestone is the creation of the web application which is broken up into a multitude of minor tasks, some of these being the log in feature and account feature. This part of the project will be started around the end of November and have some bare prototypes ready for the mid-point presentation, and plan to have many of the features working in March, which gives time to test and debug any problems that might arise before the end of the project in May. The big task with this part of the project is getting it to correctly link to the webpage and also link to the database too and most of the linking will be done in this part of the project as this is where all the work is run through.

The next major part of this project is the database that will hold the accounts and also data for the app. This will be done with MySQL. The timeline for this part of the project is to begin working on it around the start of December and be able to lay out the code backdrop for the mid-point presentation and then to have everything done for the end of February while still ensuring the code is clean and bug free all the way till the end of the project.

8.0 Testing

The testing of the project will be conducted by myself throughout the development process of the code, running the code and documenting the expected results and the actual results that was given. At the end of the process, I will then set up certain situations to run the application to ensure the code runs as expected and that there are no blind errors that I didn't fully run through in the code.

My basic expectation for the code is that the website is interactive and that it can be easily accessed, the code I expect to run the certain classes without error and run properly and for the database I expect it to be accessed by the code and the data in it is being recorded and viewed when commanded.

6.2. Reflective Journals

Supervision & Reflection Template

Student Name	Nathan Savage
Student Number	18395306
Course	BSH in Computing (Cyber Security)
Supervisor	Rohit Verma

Month: October

What?

Reflect on what has happened in your project this month?

For this month for my project, I mainly focused on my proposal ensuring that this and my project pitch video are of a good standard and that my project idea is approved so I can properly focus on the project.

So What?

Consider what that meant for your project progress. What were your successes? What challenges still remain?

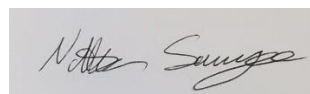
For my project progress it meant that I didn't begin working on the project directly till after I got approval which wasn't until late in the month. I was able to complete the proposal and worked on planning my project as I found out my project was approved.

Now What?

What can you do to address outstanding challenges?

The only thing I can do to address the challenges is to begin working on the project proper next month.

Student Signature



Supervision & Reflection Template

Student Name	Nathan Savage
Student Number	18395306
Course	BSh in Computing (Cyber Security)
Supervisor	Rohit Verma

Month: November

What?

Reflect on what has happened in your project this month?

This month I was able to properly start working on my project due to it being approved and began mainly working on documentation with the beginnings of the code for the web page element of my project.

So What?

Consider what that meant for your project progress. What were your successes? What challenges still remain?

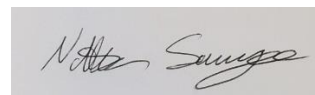
For my project progress, I was able to design a wireframe, a workflow chart, and a use case diagram to begin the documentation of the project. With these diagrams done, it gives me a strong basis to work of as I have visualised how the UI will look, how the users are meant to interact with the application and how the application runs and how the code is meant to run and work. The code I have begun working on is the make-up of the web page, this leaves the main code and database code to start which is still a lot of work to be done. Also next month is the Mid-Point deadline of my project and I will need to create a video describing what work I've done and my full idea for the project.

Now What?

What can you do to address outstanding challenges?

For the next month, I will be submitting work for the Mid-Point Implementation, Documentation and Video Presentation so I will be working to fulfil these requirements such as clean up my project proposal, fill in more of the documentation and have more code done with the web page close to finished and the main code being started.

Student Signature



Supervision & Reflection Template

Student Name	Nathan Savage
Student Number	18395306
Course	BSh in Computing (Cyber Security)
Supervisor	Rohit Verma

Month: December

What?

Reflect on what has happened in your project this month?

This month for my project I focused mainly on the mid-point presentation and ensuring all the code I had all the requirements done, such as the code, the documentation, and the video for the presentation. I made big strides in the progress of the application with getting some early versions of the main application in.

So What?

Consider what that meant for your project progress. What were your successes? What challenges still remain?

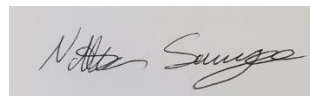
This month is a big step in the project as it is the half-way point and I needed to have a portion of the project working now. I was able to get a working site and code that showed me what my final project will entail and more, I was able to get the database linked to the application meaning I could run some queries through the app and get a proper response from it. Some challenges that still remain include tidying up the web pages and making them look better and to fully implement the working version of the main application feature.

Now What?

What can you do to address outstanding challenges?

To tidy the website I must decide on the final layout and uses for the webpage. The main application feature needs some further in detail work and also to add a more personal opinion touch on the CVE such as a scoring system.

Student Signature



Supervision & Reflection Template

Student Name	Nathan Savage
Student Number	18395306
Course	BSH in Computing (Cyber Security)
Supervisor	Rohit Verma

Month: January

What?

Reflect on what has happened in your project this month?

This was a quiet month of project work, I have been waiting to get my feedback from my mid-point submission. I have however worked on some of the documentation expanding some aspects and also working on more diagrams and charts to visualize some working aspects of the project.

So What?

Consider what that meant for your project progress. What were your successes? What challenges still remain?

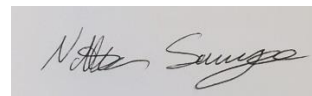
The work done this month is part of the process and part of the documentation is key to showing the thinking behind parts of the project and to visualise for myself the process that I want to provide. The challenges still include the work on the visuals of the web and further the aspects of the application.

Now What?

What can you do to address outstanding challenges?

I want to wait for my feedback before I work on some aspects of the project. I don't want to work on something only to be told it needs to be changed.

Student Signature



Supervision & Reflection Template

Student Name	Nathan Savage
Student Number	18395306
Course	BSh in Computing (Cyber Security)
Supervisor	Rohit Verma

Month: February

What?

Reflect on what has happened in your project this month?

I have made a big mistake in the explanation of my project I have been saying the main aspect of the project is a scanner when that is furthest from what I really want. I have to rectify this in my documentation and also implement some backend and simplify the database.

So What?

Consider what that meant for your project progress. What were your successes? What challenges still remain?

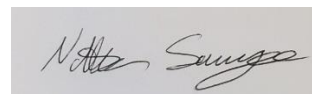
This month was an important month as it was a month where I got some proper feedback on how my project was going, I had been describing the idea and project wrong and had to rectify this in my documentation and ensure moving forward I would be using the correct terminology when addressing my project. I was also able to start on a gleaming emission for my project at this point when I began work on the backend for my project and properly add a feature that would secure the data in some way. I also began to re-arrange the way my database was set up to properly reflect how I wanted the data to be given to the user, this will make it much easier and cleaner process to view and use when it is done. As a whole this month gave me a proper idea as to what was still left to be done and the time, I have left to do it.

Now What?

What can you do to address outstanding challenges?

I have created a proper project plan which I hope will allow me to visualise the work that is left to be done and be able to work at it in a more efficient way. I have begun using Python to build my backend and learning more about the language and build on for my project. With this I will continue to build on my database as well.

Student Signature



Supervision & Reflection Template

Student Name	Nathan Savage
Student Number	18395306
Course	BSh in Computing (Cyber Security)
Supervisor	Rohit Verma

Month: March

What?

Reflect on what has happened in your project this month?

This month in my project, major progress has happened with my backend code to the point where it is mostly set up, however I have run into issues in the code running correctly. I have rearranged my database in a more suitable way, I have also begun working and research implementing a web scraper to be used to gather relevant data.

So What?

Consider what that meant for your project progress. What were your successes? What challenges still remain?

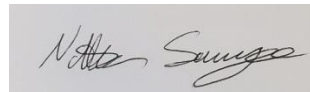
This month has seen some major progress in regard to the workings of the backend of the project with the progress made, however the issues have run into since makes testing and progress difficult as I am getting POST errors meaning the code isn't running with little other information given. This needs to be rectified to continue the project. I have made progress in other areas however without fixing the backend issues little overall progress can be made in the project.

Now What?

What can you do to address outstanding challenges?

I must focus on the problem with the backend not running properly as the project cannot work without it being rectified.

Student Signature



Supervision & Reflection Template

Student Name	Nathan Savage
Student Number	18395306
Course	BSh in Computing (Cyber Security)
Supervisor	Rohit Verma

Month: April

What?

Reflect on what has happened in your project this month?

This month had some major aspect change, I decided to restart the code for my project due to the original code I was working on getting too messy and not working the way I desired. With starting fresh, I already knew what I wanted to achieve and caught up to where I was already. I have implemented the MitreCVE API to help read and find CVEs this is a major aspect of the project.

So What?

Consider what that meant for your project progress. What were your successes? What challenges still remain?

The major success is restarting as with the clean new project I took knowledge from my first iteration and what worked with that and what didn't work I knew not to include. This shift means there is a major time crunch as to getting the project done, however I believe with the time remaining I will implement my scoring system and more profile features.

Now What?

What can you do to address outstanding challenges?

For the final two weeks of the project I plan on working to complete the code and also update some of my use cases and requirements as well as finishing the documentation and finally filming the presentation video.

Student Signature

