

Passroclus – Educational Website and Browser Add-On

**Coderona - Stuart Brown, Peter Captain, Tia
Cotton, Catriona Kirkwood, Roderick Rozalina**

CMP311: Professional Project Development & Delivery

BSc Ethical Hacking – Year 3

2020/21

TEAM MEMBER	SECTION
CATRIONA KIRKWOOD	INTRODUCTION
TIA COTTON	METHOD (API)
PETER CAPTAIN	METHOD (WEB APPLICATION AND BROWSER ADD-ON)
RODERICK ROZALINA	RESULTS
STUART BROWN	DISCUSSION

Note that Information contained in this document is for educational purposes.

CONTENTS

1	Introduction	3
1.1	Background (CK).....	3
1.1.1	Background to Problem	3
1.1.2	Project Brief	5
1.2	Aim (CK).....	5
2	Method	6
	METHOD (TC + PC).....	6
2.1	API (TC).....	6
2.1.1	Flask/API	6
2.1.2	Heroku	8
2.1.3	GitHub.....	9
2.2	User Interaction and Front-end Systems (PC)	10
2.2.1	Splash-Screen (PC)	10
2.2.2	Web Application (PC).....	11
2.2.3	Browser Add-On for Chromium based Browsers (PC).....	11
2.2.4	User Interaction (PC)	12
2.2.5	Feedback to User (PC)	12
2.2.6	Passphrase Generation and Educational Aspect (PC)	13
3	Results.....	15
3.1	Notable points (RR).....	15
4	Discussion	17
	References	18
	Appendices.....	19
	Appendix A – Executable Code – app.py (Tia).....	19
	Appendix B – User Manual (Peter).....	21
	Appendix C - Deliverables & Requirements	26
	Appendix D – Minutes (Catriona).....	28
	Appendix E – C++ Code	43
	Appendix F - Results from Website Security Evaluation (Roderick)	44

1 INTRODUCTION

1.1 BACKGROUND (CK)

1.1.1 Background to Problem

In late 2020, the National Cyber Security Centre (NCSC) published its annual review which covered the most important cyber security issues faced by the UK in 2020. The report stated that the majority of cyber threats to the public are 'low-sophistication attacks which can be prevented with just a few actions' (NCSC, 2020, p.64). It went on to say that the use of very simple, easy to crack passwords such as '123456' was common among hacking victims and that not enough people were making the required changes in order to avoid falling victim to these attacks. One of the mitigating steps recommended by the NCSC was for individuals to use strong passwords comprised of a combination of words as these are more difficult to crack using unsophisticated brute force techniques.

The pandemic has resulted in a rapid increase in online activity due to studying and working from home. Malicious individuals have used this to their advantage; Beaming recently reported that cyber-attacks on UK businesses reached an all-time high in the first quarter of 2021 as shown in figure 1, representing an 11% increase on the same period of 2020.

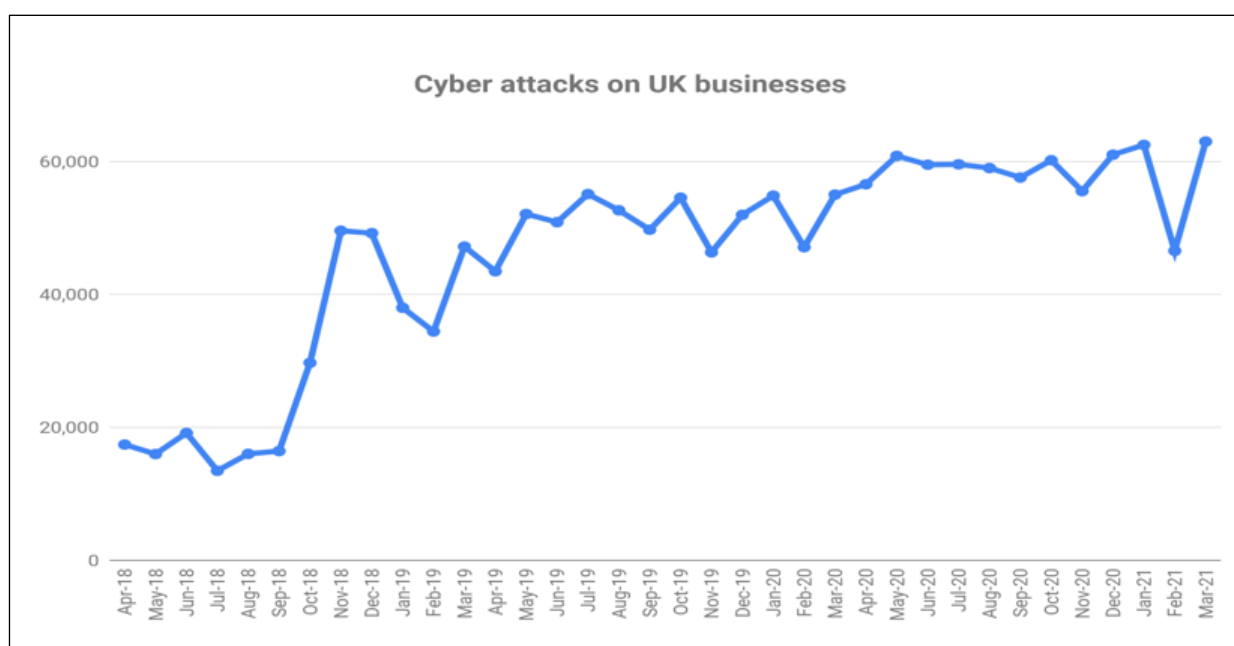


Figure 1: Average number of cyber-attacks per UK business (Beaming, 2021).

The 2021 Cyber Security Breaches Survey published by the Department for Digital, Culture, Media and Sport (DCMS) stated that the percentage of further education colleges that have identified a breach or attack in the previous twelve months was 75% as shown in figure 2. Of that 75%, 74% reported a negative impact such as having to divert staff resources to handle the breach, and disruption to staff and students. The sample size for universities was considered too small to be reliable, however, the university data also showed a high number of attacks experienced by these institutions.

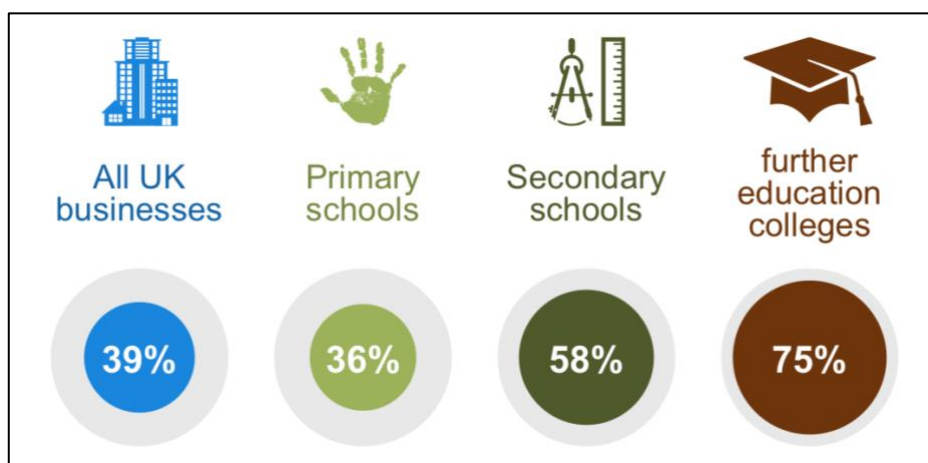


Figure 2: Percentage of organisations that have identified breaches or attacks in the last 12 months (DCMS, 2021).

In early 2021 there was a further increase in cyber-attacks in the UK education sector, this led to the NCSC releasing specific guidance for the sector in March 2021 with the aim of increasing resistance to such attacks. As in their 2020 annual review, the NCSC reiterated the importance of strong passwords in the defence against hackers. Given the high rate of cyber-attacks and general lack of knowledge and poor password behaviour among the British public, it is important that good password behaviour is encouraged and facilitated.

There are some existing password tools on the market that aim to address the problem of poor password behaviour in the general public, a summary of the functionality provided by these tools can be seen in table 1. The Kaspersky password tool incorporates a breach checker, however this makes use of the 'Have I Been Pwned' password breach checker which is a very impressive tool, but means that Kaspersky does not have full control over the breach checking functionality. Although some of these tools are very good, none of them offer all four functionality attributes; research conducted by Coderona did not identify any tools that provided the full range of functionality. This means that users have to look elsewhere for the additional functionality that they may require. As previously mentioned, the NCSC has stated that not enough people are making the necessary changes to improve their password behaviour, it would therefore be preferable to have a tool that provides all functionality in one place, maximising convenience for the user. There is a gap in the market for a password tool that provides greater functionality than the current market offerings and does not use other existing tools to provide that functionality.

Table 1: Comparison of functionality provided by password tools currently available.

	How Secure Is My password?	My1Login	Kaspersky	The Password Meter	Comparitech
Strength Check	✓	✓	✓	✓	✓
Breach Check			✓		
Password Generator					✓
Educational Information	✓	✓	✓		✓

1.1.2 Project Brief

Dr. Ethan Bayne of Abertay University contacted Coderona in late 2020 with a view to developing a tool to address the public knowledge gap in the area of password creation. The product was intended to meet the following specifications:

- Check a password for common flaws
- Check if a password has been breached
- Help users to create better passwords using current best practices
- Educate users on the importance of using strong passwords
- Be intuitive and usable
- Handle data securely and locally

1.2 Aim (CK)

The aim of this project was to create an all-in-one password tool named Passroclus that meets the requirements outlined in the project brief (section 1.1.2) and comes in the form of a web application and browser add-on. The purpose of the product is to improve the knowledge of password creation and best practice for users of the Abertay University system, and thus reduce the vulnerability of the system to brute-force attacks and the negative impacts associated with successful attacks. The following objectives outline the work that was involved in the creation of the Passroclus tool:

- Development of an API
- Creation of a browser add-on
- Creation of a website
- Development of code that calculates the strength of a password
- Development of code that checks a password against a list of breached passwords
- Development of code that generates a new password

Coderona focused on the end user throughout the project; Passroclus was designed to be highly intuitive to use. The team utilised technologies that are widely accessible on many devices and web browsers to ensure that all potential users are able to access the product. Other work included regular testing of the product during the development phase to ensure that it is fully functional and free from errors. Consideration was given to the possibility of changes such as:

- NCSC password advice may change
- Abertay University may update their minimum password requirements
- Advice on calculating password strength may change
- New password breach lists may be released

Since none of the functionality is provided by a third-party tool, the Passroclus code is fully customisable to ensure that the aim of providing current best practice advice can be maintained, regardless of whether advice changes in the future. Use of the tool should result in users improving their password knowledge and behaviour, something that is essential for Abertay University given the high level of cyber-attacks in the education sector and lack of knowledge in the area of password best practice.

2 METHOD

Colour Code = TC is blue, and PC is dark green

METHOD (TC + PC)

The Passroclus tool was created with the use of Flask to build a custom API, that was then hosted on Heroku through GitHub. Passroclus has a responsive web application as well as a browser add-on that mirrors the web application. In this section, Tia will discuss the development of the API – going into depth about Heroku, GitHub and the API itself. Peter will discuss the web application and the browser add-on, how it works and how it was built.

2.1 API (TC)

The API (Application Programming Interface) has been built with flask, Heroku and GitHub. Due to technical difficulties with the university servers and technologies, Coderona looked for similar alternatives that were cost-free and technically suited for the project.

The diagram below in figure three, is a simple overview of the API ecosystem. Within the ecosystem, GitHub provided the resources and materials for Heroku to build and deploy the API. Heroku hosted the API as an application within the cloud and responds to application requests by the user.

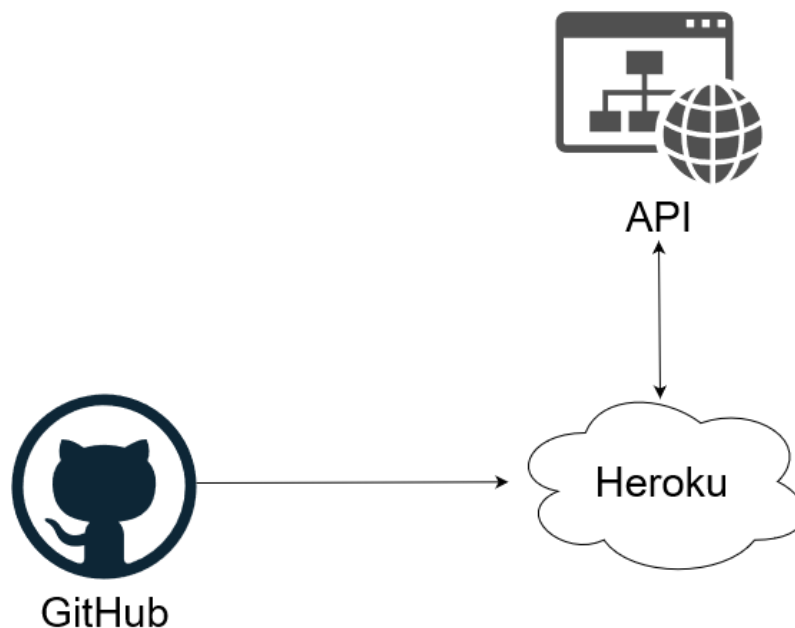


Figure 3 - Ecosystem of the Passroclus application

2.1.1 Flask/API

The API was built using Flask, which is a web framework built in python. Flask was used instead of other alternatives such as Django, as it was more suited to API's and rapid development of a single application. The full source code for the API can be found at appendix A. The python file contained the code for each of the functions as well as the index page for the web application. There were two main functions within the API, which were securityChecker and generator.

The two functions were originally built in C++, however due to technical issues they were not able to be used. The libraries that were chosen to merge the C++ and python together did not work as expected, so the code was used as a reference for the functions below. The C++ code can be found in the GitHub repository.

The securityChecker function taken the user's entered password and calculated the overall entropy score and checked for possible breaches. The securityChecker function would count how many ASCII characters, numbers and lower and uppercase characters are within the password. They were then used to calculate the entropy of the password, the process of this can be seen below in figure 4.

```
@app.route('/securityChecker/<passWord>')
def securityChecker(passWord):
    symbols = [' ', '!', '#', '$', '%', '&', "'", '?', '@']
    lowercase = 0
    uppercase = 0
    num = 0
    ASCII = 0
    uniqueChar = 0
    rating = ""

    length = len(passWord)

    if (len(passWord) >= 1):
        for i in passWord:
            if (i.islower()):
                lowercase += 1
                uniqueChar = uniqueChar + 26
            if (i.isupper()):
                uppercase += 1
                uniqueChar = uniqueChar + 26
            if (i.isdigit()):
                num += 1
                uniqueChar = uniqueChar + 10
            if (i == symbols):
                ASCII += 1
                uniqueChar = uniqueChar + 30
        entropy = math.log2(uniqueChar**length)
```

Figure 4 - Excerpt of securityChecker function in the API

When the entropy has been calculated, the breach checker function is run to make sure that the password is secure. If it has been breached the overall entropy is disregarded and the password is weak, with the 'score' being red. If the password has not been breached, the 'score' will be changed according on the value of the calculated entropy. The scoring system can be seen below in figure five.

```

if (entropy <= 27):
    rating = "Your passphrase is very weak, consider using our passphrase generator to create a new one"
    score = "red"
elif (entropy <= 35):
    rating = "Your passphrase is weak, consider using our passphrase generator to create a new one"
    score = "red"
elif (entropy <= 59):
    rating = "Your passphrase is reasonable"
    score = "orange"
elif (entropy <= 127):
    rating = "Your passphrase is strong!"
    score = "green"
elif (entropy >= 128):
    rating = "Your passphrase is very strong, well done!"
    score = "green"

result = open('ncscTop100k.txt', 'r')
if (passWord in result.read()):
    breachedPassword = "This password has been found in a breach, we suggest changing this password anywhere you use it.";
    breached = "true"
    overallScore = "#eb5160"
else:
    breachedPassword = "This password was not found in a breach."
    breached = "false"
    if (breached == "false" and score == "red"):
        overallScore = "#eb5160"
    elif (breached == "false" and score == "orange"):
        overallScore = "#008967"
    elif (breached == "false" and score == "green"):
        overallScore = "#688f71"

```

Figure 5 - Scoring in the securityChecker function

The generator function – which can be seen in figure six - taken an entered number from the user which determined how many words they would like to have in their passphrase. The API generated used this number to grab the set number of random words from a file containing around two thousand, five hundred English dictionary words. These words were then displayed to the user as their new passphrase.

To randomly choose the words, the randomWords file was loaded into an array. When all the words were in the array, the random function was used to generate a random position in the array. The word that was at the position in the array was then added to the passphrase variable. This process was within a loop that ran for the number the user entered into the generator at the start.

```

@app.route('/generator/<length>')
def generator(length):
    length = int(length)
    textFile = open('randomWords.txt', 'r')
    textFile = textFile.read()
    words = list(map(str, textFile.split()))
    passphrase = []

    for i in range(length):
        temp = int(random.random()*len(words))
        passphrase.append(words[temp])

    return render_template('generator.html', passphrase = passphrase)

```

Figure 6 - generator function within the API

2.1.2 Heroku

Due to issues that were encountered early in the development stage with the university servers, Heroku was used instead to host the application within the cloud. Heroku is a container-based platform-as-a-service tool that allowed Coderona to deploy and manage Passroclus on the cloud.

Coderona simply had to connect the Passroclus github to Heroku and configure the dyno to have the application hosted on the cloud.

Passroclus was placed into a 'dyno' container that allowed Passroclus to work as an application within the cloud. Gunicorn was used as a HTTP server and was used in Heroku to host the Passroclus application. The dyno that was used for Passroclus was the free web dyno, which is built for experimental applications, see figure four for an example of this.

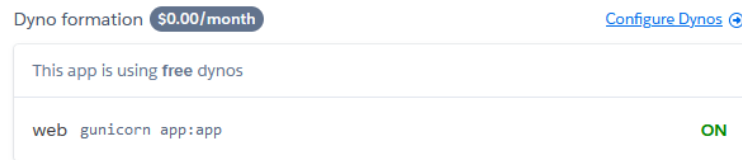


Figure 7 - Dyno used for passroclus

With Heroku being linked to the GitHub repository of the Passroclus application, the most up-to-date version of the tool was always available to the user – this is demonstrated in figure eight. Automatic deploys were applied after the final version of the prototype had been built, but manual deploys were used throughout the duration of the development phase.

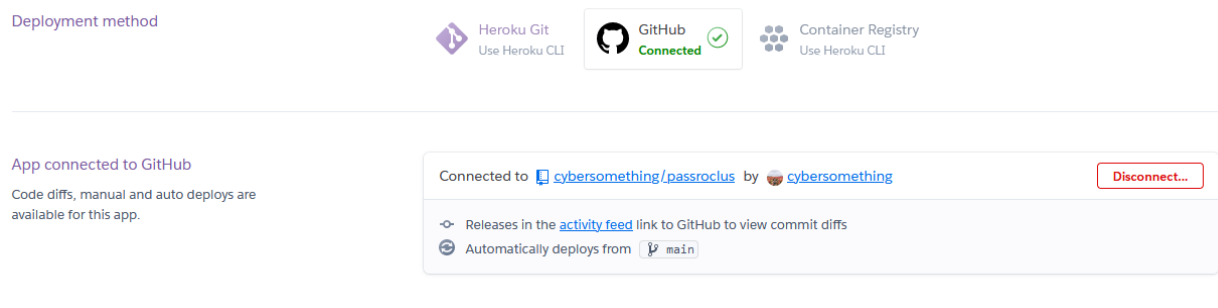


Figure 8 - Deployment Selection for Passroclus in Heroku

2.1.3 GitHub

The application's resources were on GitHub, rather than having to manually transfer material to a server – Heroku automatically deployed the newest version of the application from GitHub. This meant that the most current version of the application is always available to a user.

As Passroclus was being hosted on Heroku, GitHub was used to store the code for the API and other relevant resources required for the application. These resources included the text files that were used for the securityChecker and generator functions. The requirement and runtime files that Heroku needs to deploy and host the Passroclus application on the cloud. The repository can be [found here at cybersomething/passroclus](https://github.com/cybersomething/passroclus).

The requirement file in GitHub contains the dependencies that Flask requires to function effectively. These dependencies and their required versions can be seen in figure nine. There were eight dependencies in total, including the Flask library. The dependencies that are mainly used within Passroclus were Gunicorn, Jinja, Werkzeug and request.

```
1 Flask==1.1.1
2 gunicorn==19.9.0
3 Click==7.0
4 itsdangerous==1.1.0
5 Jinja2>=2.11.3
6 MarkupSafe==1.1.1
7 Werkzeug==0.15.6
8 requests==2.20.1
```

Figure 9 - Contents of requirements.txt

Jinja was used to display the API results within the HTML templates, whilst Werkzeug was used for the API to communicate effectively with the web application. Requests allowed the API to accept requests from the application such as the entered password and the requested number of words in the passphrase.

2.2 USER INTERACTION AND FRONT-END SYSTEMS (PC)

2.2.1 Splash-Screen (PC)

Coderona's solution, "Passroclus", is seen by the user in our front - end web application and browser extension. This software uses the Passroclus API, which makes use of the technologies listed previously. Both are presented using the same format, as seen in figure 10. The Passroclus API is essentially a mixture of 3 utilities which the average user can navigate through use of Passroclus' splash screen. The splash screen itself allows the user to choose if they want to analyse a password for strength and check if it has been involved as part of a breach. It also allows the user to select the option to generate a secure passphrase, as well as learn about passwords.

The image shows a dark-themed splash screen for 'Passroclus'. At the top, it says 'Welcome to Passroclus.' in white. Below that, it says 'Enter the password to be checked.' in white. There is a white input field with the placeholder text 'Enter your password here'. Below the input field is a blue button with white text that says 'Check Password'. Further down, it says 'Or, create a new passphrase below.' in white. Below this is another blue button with white text that says 'Create a passphrase!'. At the bottom, it says 'Want to read the FAQ's?' in white. Below this is a light blue button with dark blue text that says 'Go to FAQ's'.

Figure 10 - Passroclus' splash screen

The splash screen is designed to be accessible and welcoming. Text is displayed in a large, easy to read font, buttons and input-boxes where the user can enter their password are clearly defined and the entire utility maintains this layout in both the types of chromium based browsers the utility was designed for, (Google Chrome and Microsoft Edge) Although the current look of Passroclus is subject to change upon deployment as per the clients needs and desires, thought was given as to how Passroclus should look. It has been developed initially using the Passroclus

“theme” colours, of blues and greens. These were chosen as blue is associated with representing harmony and intelligence, whereas green has connotations of safety and hope. The two colours harmonise and together they help make the Passroclus utility look more professional, calming and confident of its ability to “oust bad passwords from users”.

2.2.2 Web Application (PC)

Passroclus was originally designed for use on its own website. This is where testing and refinement of the final design was completed before presentation to the client. Passroclus is available on any web browser by following the link, however it was originally displayed only in Google Chrome and Microsoft Edge, as that is where the associated browser Add-On would be used. As a result, Passroclus can be accessed through the world-wide-web, even on smartphones, as seen in Figure 11.

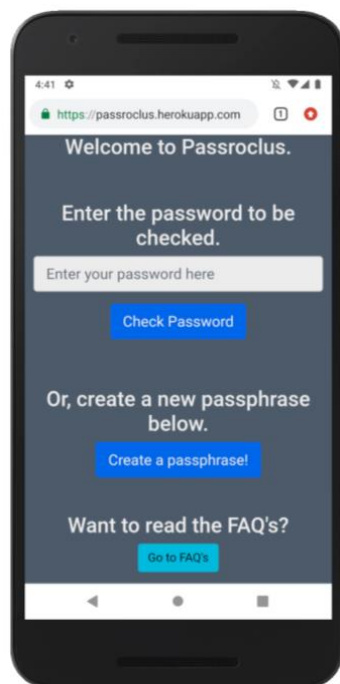


Figure 11 - Passroclus is also available on Mobile devices

2.2.3 Browser Add-On for Chromium based Browsers (PC)

The browser add-on is essentially a “lite” version of the website. As such, it has a lot in common with the website, such as the layout, theme and the reliance on the Passroclus API to perform tasks. Uniquely however, is that the Browser add on is designed to offer the full range of services provided by the website in an easy to use package that can be located in a user toolbar when browsing the internet. This enables Passroclus to maintain a presence when the user needs it the most: creating a password on another website. The browser add on can be seen in Figure 12.

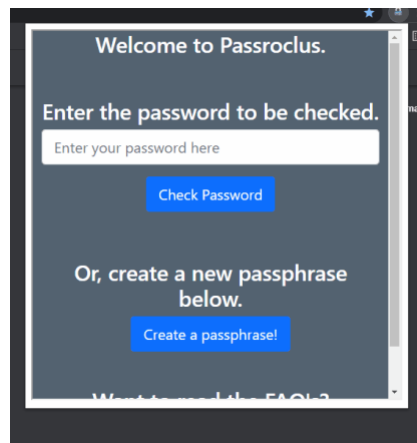


Figure 12- The Browser Add-On, a portable version of the web application

2.2.4 User Interaction (PC)

The user interface is designed to be as clean as possible, to help improve navigation for people who use it. On both the web application and browser add on the user can access Passroclus for a Password Strength check, which combines the password strength and breach checkers that were developed originally in C++ as mentioned earlier.

It should be noted that should the server Passroclus is run from change, such as on launch, it may be possible to re-use these much “heavier” programs that have the capabilities to check for a much wider array of attributes associated with secure passwords as well as search through a much larger database of broken passwords. These C++ programs can be found in the appendix E. This is not strictly necessary, Passroclus will still perform to a high level with the utilities it uses currently. The use case for Passroclus is derived from a scenario in which a user who wishes to check their password as well as learn more about good password security needs an all-in-one utility to do so. As such, Passroclus’ interface has three main components. After immediately opening up the web browser or the add-on, the user is welcomed by the splash screen. The first utility to get the users attention, as they look down the page, is the password strength checking functionality where the user can paste in their password and have it checked by the entropy and breach checker. To make use of this functionality, the user enters their password and receives feedback using the “traffic-light system”.

2.2.5 Feedback to User (PC)

Upon checking their password, Passroclus will feedback its findings to the user through the feedback page, which is immediately opened up as soon as the API has finished making calls. The user can then read what Passroclus “thought” about the input and be educated as to any further action that needs to be taken. There are several ways that feedback can look, however all types of feedback will detail the passwords entropy score, if the password was involved in a breach, an explanation of what entropy is as well as link to find out more about entropy.

If a password was identified as part of a breach the traffic light colour the background becomes will always be red, even if the overall entropy of that password is very good (see Figure 13). This is to help sway off the user from using the password. If the password was not involved in a breach then the background colour can become one of three colours in a typical traffic light. It will become red if the entropy is poor, amber if the entropy is average and green if the entropy is good.



Figure 13 - Using a strong password, even if it has been involved by a breach, will always result in a red.

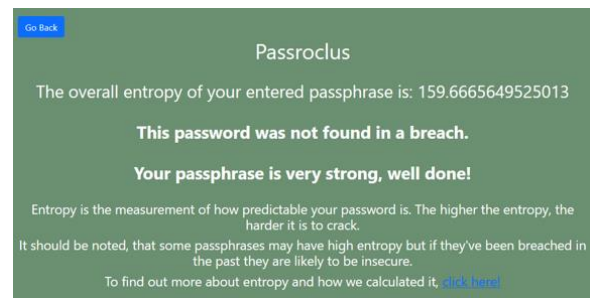


Figure 14 - A strong, not breached password will cause the background to become green



Figure 15 - A password of average strength will be labelled as "reasonable" and change the background to amber.

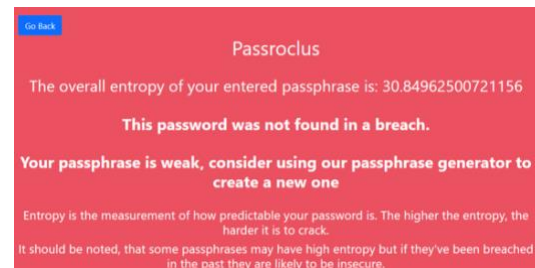


Figure 16 - A password of poor strength will be labelled as "weak" and cause the background to become red.

2.2.6 Passphrase Generation and Educational Aspect (PC)

If the user wishes to instead forego creating and using Passroclus to check their password, they can make use of Passroclus' passphrase generation tool as well as the FAQ section. These can be seen in Figures 17 and 18. By accessing passphrase generation, the user can request a passphrase of 2 - 6 words that are assembled randomly by the passphrase generator. The passphrase that is generated can then be assembled and fed into the strength checker just to be sure, however the phrase that is generated, due to its length, is likely to be very secure regardless.

If the user wishes to know more about how Passroclus works, they can access the FAQ's page (figure 18). This explains how the strength of the password is assessed as well as why a passphrase is better than a password.

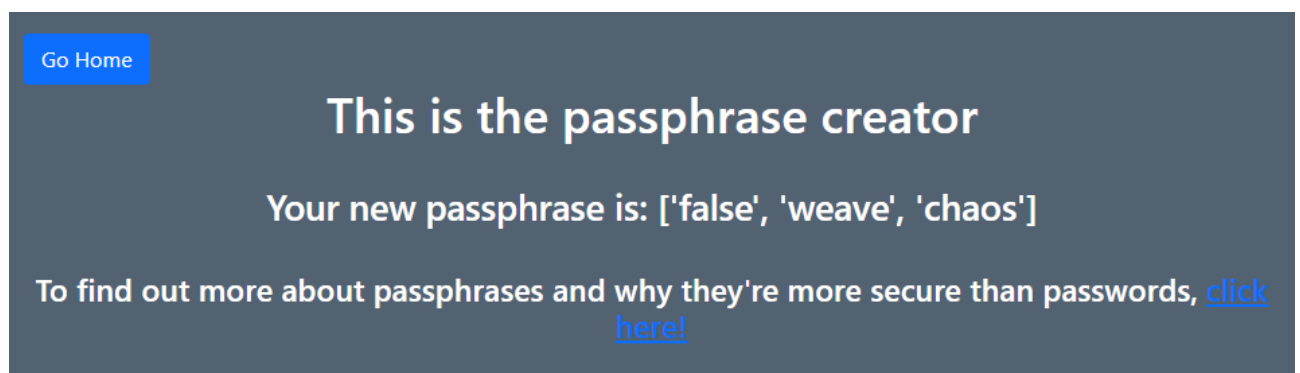


Figure 17 - Passphrase Generator

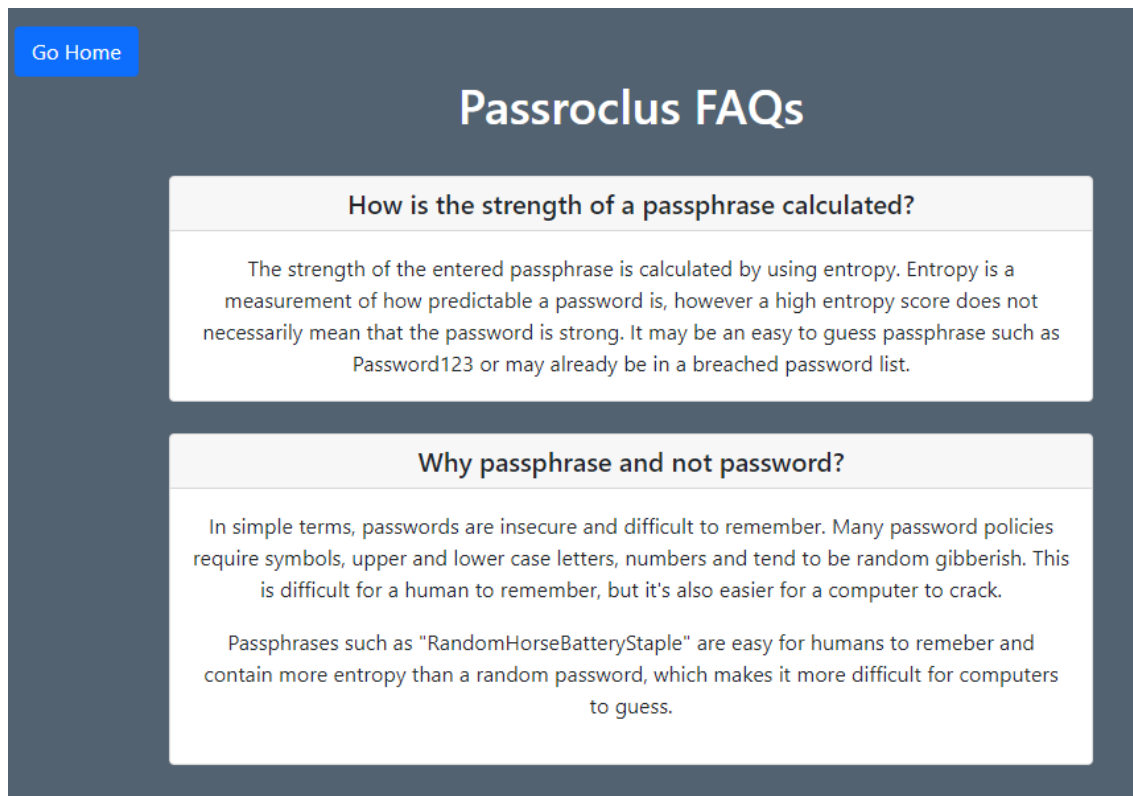


Figure 18 - the FAQ Page.

3 RESULTS

3.1 NOTABLE POINTS (RR)

Extensive research was conducted for us to create a product that would meet the needs and requirements of today's password challenges. Our goal was to ensure that users are aware of the dangers of creating weak passwords for a website. That problem was then implemented into a solution that we believe would be needed to educate users on the password choices and provide them with versatile passwords that meets not only the length, but complexity as well. Our aims were met by making Passroclus accessible on mobile, as well as a web application/ browser add on to reach out to more users across the technological landscape. Furthermore, with further implementation, it can be easily be expanded and applied on other web browsers. The technologies used ensure that services are as cost efficient as possible, while delivering a service that performs despite the circumstances.

Data/Performance Indicators.

Due to the current pandemic, user data could not have been generated for evaluation purposes. Instead our focus mainly aimed on the performance and response time of the add-on. During the prototyping phase, performance had been adjusted based on the technologies used and the final product ensures that such speed performance requirements were met.

Usability

We've decided with the help of With the Web Accessibility evaluation Tool, (WAVE) we can make the website easily accessible, including users with vision disabilities. By this, we can focus on points from the analysis to make our web application user friendly along with the flow of the website. In Appendix F figure 1, an analysis of the website was made to see what errors or warnings appeared and what can be reconfigured to better the usability for users. The evaluation contained 1 error, 1 contrast errors, 1 alert that needed to be addressed. The errors noted are as follows:

- Error, Missing form Label.
- Low Contrast Between text and background.
- Lack of Page regions.

WAVE then returned feedback on what the issues were and gave solutions to the issues. These issues indicated were rather small and could be resolved instantly. Aside from the minor errors that were present, WAVE also indicated us on the plus points from our website. The HTML language was properly indicated within the file, along with proper use of structural elements to differentiate between header sizes where necessary. This meets the aim of helping as many people as possible.

The user interfaces were designed initially via a wireframe concept to puzzle out the order/flow and accessibility of the website(Appendix F Figure 1-A). The aim was to ensure that the navigation for users be as smooth and straightforward as possible. The traffic-light system that was implemented to properly indicate to users whether their passwords need to be reconsidered,

adjusted and perfected according to their inputs. Feedback given in the bottom of the page educated the user on their choices and explains the reasoning to why such feedback was given.

Users can always rest assured that updates will be implemented for both the browser add-on as well as the web application itself. This is to keep our product up to date with modern trends and new breaches. This way affected users can always check as soon as the information is released.

Vulnerabilities found.

While the website was designed with security in mind, a scan was performed on the website to indicate any potential vulnerabilities that may be present within the pages itself. The tool used for this scan was the open source web vulnerability OWASP Zap. In Appendix F, the results of the scans can be seen along with the vulnerabilities presented. This was done during the prototyping phase to see what issues can be rectified from the get-go. In this case, along with the WAVE tool, we've managed to eliminate potential attack vectors from our website.

Noted Alerts.

The results came back with 4 minor alerts that can be easily configurable prior to product release. While nothing is perfect upon development, the results came back with the following:

- X-frame Option Not set.
- No Anti-CSRF Tokens.
- Incomplete/No Cache control.
- X-Content Type Options Header Missing.

Minor alerts can be easily rectified by reading solutions presented within the scan itself to ensure that the site functions according to modern web standards. With this, Coderona strived to reduce the amount of broken code within the website and focused on technologies that would be secure upon deployment.

Server Communication.

Security measure that was discussed during the planning phase was analysing the security of the communications between server and client. During breach checking, the list that is used to withdraw information about the breached passwords has been reduced. This means that if a user states the password that needed to be checked with the breacher, if said password was breached, the overall list is randomized to not reveal the breached password. Example of this is If a user entered a common password containing the letter "A", the list then gives every password containing the letter "A" in it. Therefore, when searching, the common elements withdrawn from every password is the letter. No further clues are then given to what the actual password might be. This prevents malicious users attempting to gain information about the passwords used in breaches.

We at Coderona believe that with Passroclus, we can educate users on developing strong password creating habits and offering solutions guided by research and modern web standards, to render simple passwords ineffective, and thus making brute forcing for passwords difficult. This, in combination with strong and effective web security measures can be effective against breach gathering and password reuse amongst the web.

4 DISCUSSION

Due to personal, mitigating circumstances, Stuart Brown was unable to complete this section. The module leader has been notified and is aware of this.

REFERENCES

Beaming. (2021). *Q1 2021 Cyber Threat Report*. [Online]. Available at: <https://www.beaming.co.uk/cyber-reports/q1-2021-cyber-threat-report/> (Accessed 20/4/21).

Comparitech. (No date). *Password Strength Test & Strong Password Generator Tool*. [Online]. Available at: <https://www.comparitech.com/privacy-security-tools/password-strength-test/#password-test-tool> (Accessed 20/4/21).

Department for Digital, Culture, Media & Sport. (2021). *Cyber Security Breaches Survey 2021*. [Online]. Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/977490/Cyber_Security_Breaches_Survey_2021_Education_Annex.pdf (Accessed 20/4/21).

Kaspersky. (No date). *Check your password*. [Online]. Available at: <https://password.kaspersky.com> (Accessed 20/4/21).

My1Login. (No date). *How secure is your password?* [Online]. Available at: <https://www.my1login.com/resources/password-strength-test/> (Accessed 20/4/21).

National Cyber Security Centre. (2020). *Annual Review 2020*. [Online] p.64. Available at: https://www.ncsc.gov.uk/annual-review/2020/docs/ncsc_2020-annual-review_s.pdf (Accessed 20/4/21).

National Cyber Security Centre. (2021). *Alert: Further targeted ransomware attacks on the UK education sector by cyber criminals*. [Online]. Available at: https://www.ncsc.gov.uk/news/alert-targeted-ransomware-attacks-on-uk-education-sector#section_6 (Accessed 20/4/21).

Security.org. (No date). *How Secure Is My Password?* [Online]. Available at: <https://www.security.org/how-secure-is-my-password/> (Accessed 20/4/21).

The Password Meter. (No date). *The Password Meter*. [Online]. Available at: <http://www.passwordmeter.com> (Accessed 20/4/21).

APPENDIX A – EXECUTABLE CODE – APP.PY (TIA)

```
# app.py
#Most current working version as of 13/04/2021
import os
import subprocess
import re
import random
import json
import math
from flask import Flask, request, jsonify, render_template, url_for, redirect, flash
app = Flask(__name__)
@app.route('/securityCheckerRedirect', methods = ['POST', 'GET'])
def securityCheckerRedirect():
    if request.method == 'POST':
        return redirect (url_for('securityChecker'))
@app.route('/securityChecker/<passWord>')
def securityChecker(passWord):
    symbols = [' ', '!', '#', '$', '%', '&', '"', '?', '@']
    lowercase = 0
    uppercase = 0
    num = 0
    ASCII = 0
    uniqueChar = 0
    rating = ""

    length = len(passWord)

    if (len(passWord) >= 1):
        for i in passWord:
            if (i.islower()):
                lowercase += 1
                uniqueChar = uniqueChar + 26
            if (i.isupper()):
                uppercase += 1
                uniqueChar = uniqueChar + 26
            if (i.isdigit()):
                num += 1
                uniqueChar = uniqueChar + 10
            if (i == symbols):
                ASCII += 1
                uniqueChar = uniqueChar + 30
        entropy = math.log2(uniqueChar**length)

    if (entropy <= 27):
        rating = "Your passphrase is very weak, consider using our passphrase generator to create a new one"
        score = "red"
    elif (entropy <= 35):
        rating = "Your passphrase is weak, consider using our passphrase generator to create a new one"
```

```

    score = "red"
elif (entropy <= 59):
    rating = "Your passphrase is reasonable"
    score = "orange"
elif (entropy <= 127):
    rating = "Your passphrase is strong!"
    score = "green"
elif (entropy >= 128):
    rating = "Your passphrase is very strong, well done!"
    score = "green"

result = open('ncscTop100k.txt', 'r')
if (passWord in result.read()):
    breachedPassword = "This password has been found in a breach, we suggest changing this password
anywhere you use it.";
    breached = "true"
    overallScore = "#eb5160"
else:
    breachedPassword = "This password was not found in a breach."
    breached = "false"
    if (breached == "false" and score == "red"):
        overallScore = "#eb5160"
    elif (breached == "false" and score == "orange"):
        overallScore = "#DDB967"
    elif (breached == "false" and score == "green"):
        overallScore = "#6B8F71"

return render_template('securityChecker.html', entropy = entropy, rating = rating, breachedPassword =
breachedPassword, overallScore = overallScore)

@app.route('/checker', methods = ['POST', 'GET'])
def checker():
    if request.method == 'POST':
        password = request.form['password']
        return redirect(url_for('securityChecker',passWord = password))
    else:
        password = request.args.get('password')
        return redirect(url_for('securityChecker',passWord = password))

@app.route('/creator', methods = ['POST', 'GET'])
def creator():
    return render_template('creator.html')
@app.route('/generatorRedirect', methods = ['POST', 'GET'])
def generatorRedirect():
    if request.method == 'POST':
        length = request.form['length']
        return redirect(url_for('generator', length = length))
    else:
        length = request.args.get('length')
        return redirect(url_for('generator', length = length))
@app.route('/generator/<length>')
def generator(length):
    length = int(length)
    textFile = open('randomWords.txt', 'r')

```

```

textFile = textFile.read()
words = list(map(str, textFile.split()))
passphrase = []

for i in range(length):
    temp = int(random.random()*len(words))
    passphrase.append(words[temp])

return render_template('generator.html', passphrase = passphrase)

# A welcome message to test our server
@app.route('/')#, methods=['POST'])
def index():
    return render_template('index.html')

@app.route('/FAQ')
def faq():
    return render_template('faq.html')

if __name__ == '__main__':
    # Threaded option to enable multiple instances for multiple user access support
    app.run(threaded=True, port = int(os.environ.get('PORT', 5000)))

```

APPENDIX B – USER MANUAL (PETER)

See the following pages

Passroclus User Guide



Passroclus

Struggling as to how to improve your password? No problem!

Passroclus is Coderona's all in one utility to help users worldwide achieve a better password!

This guide will walk you through, step by step. all of Passroclus features.

Step 1)

Open up the browser add-on on a chromium-based browser (Google or Edge), this will be located at the **top right** of the bar that the URL is displayed in. Don't have Passroclus installed? Go to <https://passroclus.herokuapp.com/> to access the website.

Once Passroclus is loaded the browser / add on should pop up and appear like this:

The image shows a dark grey rectangular window with white text. At the top, it says "Welcome to Passroclus." Below that, it says "Enter the password to be checked." There is a white input field with the placeholder text "Enter your password here". To the right of the input field is a blue button that says "Check Password". Below this, it says "Or, create a new passphrase below." There is a blue button that says "Create a passphrase!". At the bottom, it says "Want to read the FAQ's?" with a small teal button that says "Go to FAQ's".

If you want to check the strength of your password or see if it's breached, go to **Part 2**.

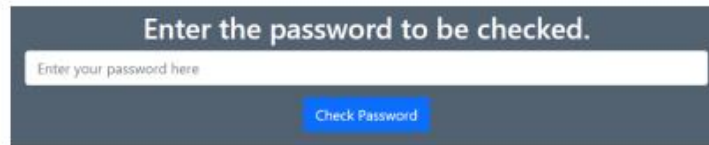
If you want to generate a passphrase, go to **Part 3**.

If you want to check the FAQs go to **Part 4**.

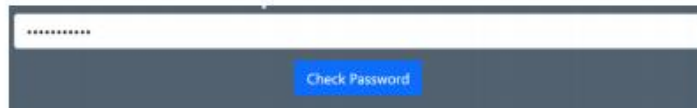
If you want to navigate the application better, go to **Part 5**.

Passroclus is a registered Coderona Product

Part 2) Strength and Breach Checker



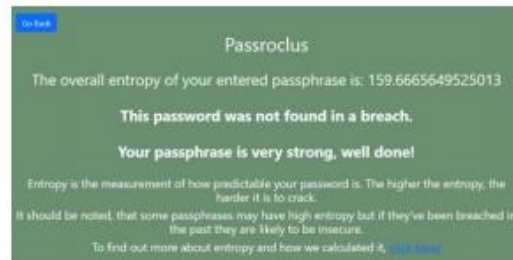
Immediately underneath the “Welcome to Passroclus” message, you will notice a large input box, as seen above. Enter the password you want to check in this box then click the button that says: “**check password**”, as shown below.



When you click this button, the screen may look like those listed below. Please refer to the instruction below the picture which looks most like yours.



RED, Breached - Go to 2a



GREEN, Not Breached - Go to 2b



AMBER, Not Breached - Go to 2c



RED, Not Breached - Go to 2d

- 2a) Your password was identified as being part of a **previous breach**. It is likely this password was leaked and tools that brute-force passwords might break this password with ease.
- 2b) Your password was identified as being **strong**, as well as not part of a previous breach. The entered password should be safe to use!
- 2c) Your password was identified as being of **average** strength. Consider adding length to the password or adding in more special characters such as * (<) ? \ ...etc.
- 2d) Your password was identified as being of **weak strength**. It is suggested you make your password longer and special characters to make it stronger.

Part 3) Create a passphrase

Or, create a new passphrase below.

Create a passphrase!

By clicking on the box that says “**Create a passphrase**” you will be taken to the passphrase generation tool, as seen below.

Passroclus - Passphrase Creator

How long would you like the password to be?:

Submit

To find out more about passphrases and the passphrase generator, [click here!](#)

You can then choose how many words you want the passphrase to consist of. You can select between 2 and 6 words to compose the passphrase. It is suggested you select:

2 if you need a quick, single use password.

3 or 4 if you plan on remembering the password.

5 or 6 if you want a password long enough that a machine will have a hard time breaching it.

Once you hit the “submit” button, the next page will list the selected words to make your passphrase with, as the example below demonstrates.

This is the passphrase creator

Your new passphrase is: ['affair', 'flour', 'proud']

To find out more about passphrases and why they're more secure than passwords, [click here!](#)

In this example you could enter your password as:

affair_flour_proud,

affairflourproud,

flouraffairproud.

However it would be better to add some special characters or numbers you remember to improve the passphrase's entropy. For example the passphrase could become :

Affair03Flour05Proud2000

or even evolve it into

@FF@1R/fl0Ur\pRoUd

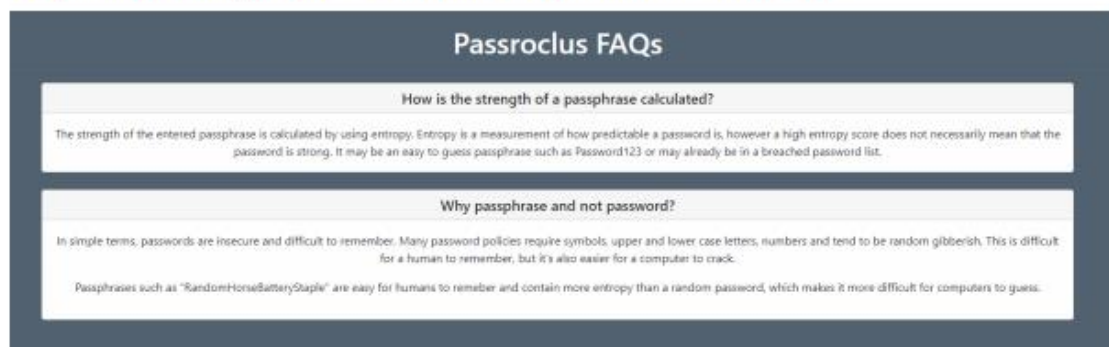
If you wish to learn more about the password, clicking “**click here**” will take you the FAQs.

re than passwords, [click here!](#)

Part 4) The FAQs

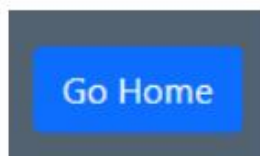


If you would like to learn about what makes a good password, then clicking on **"Go to the FAQs"**, as seen above, will take you to some useful information. Such as how Passroclus evaluates the strength of a password or why passphrases are better than passwords, as seen below

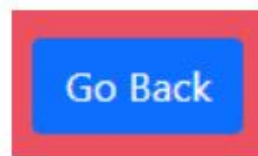


Part 5) Navigation.

At several points throughout Passroclus you should notice a button located at the top left of the screen. Hitting this button will take you home. This sign will be labelled as **"Go back"** or **"Go Home"**.



"Go Home" - as it appears in the FAQs



"Go Back" - as it appears in Strength Checker

Need further help?

Please do not hesitate to get in touch with Coderona at:

TEAM CODERONA

Passroclus_support@coderona.scot

ABERTAY UNIVERSITY

DUNDEE, DD1 1HG

Passroclus is a registered Coderona Product

Agreement Form: Project Deliverables

Group Name, Names of Team Members, and Programme	Coderona: Stuart Brown, Peter Captain, Tia Cotton, Catriona Kirkwood, Roderick Rozalina (BSc Ethical Hacking)
Subject specialist's Name (Client)	Ethan Bayne
The deliverables listed below will be submitted by the team by the due date.	
Part A deliverables	To be agreed by programme specialist and team, for example: <ul style="list-style-type: none">• API with website and browser add-on• Executable code/investigation report• User manual• Requirements Specification, signed off by the programme specialist (see overleaf)• Product Testing Report (Inc. Vulnerability and Security assessment)
Subject specialist's signature	Ethan Bayne
Team members' signatures	Stuart Brown, Peter Captain, Tia Cotton, Catriona Kirkwood, Roderick Rozalina

Agreement Form: Requirements

Group Name: Coderona

Team members (print): Stuart Brown, Peter Captain, Tia Cotton, Catriona Kirkwood, Roderick Rozalina

Project Title: Passroclus, Flaws in password data

Please refer to the attached documentation for full details on the project. The requirements are listed in Table 1. The signatures below indicate that the requirements for this project have been agreed by the project stakeholders.

Any changes to the project documentation should be made using the correct change authorisation procedure agreed with the programme specialist.

ID	List of Agreed Requirements (fill in)
1	Password Strength Checker
2	Breach Checker
3	Strong Password Creator
4	User friendly interface
5	Education of password hygiene and security

Stakeholders	Signatures	Date
Team members	Stuart Brown, Peter Captain, Tia Cotton, Catriona Kirkwood, Roderick Rozalina	15/02/2021
Programme Specialist	Ethan Bayne	2021-02-22
Client (if applicable)	Ethan Bayne	2021-02-22

APPENDIX D – MINUTES (CATRIONA)

Coderona Passroclus Project Team Meeting Week 1 (Online: Microsoft Teams) Minutes of the meeting held on 25th January 2021

Present:

Stuart Brown
Peter Captain
Tia Cotton
Catriona Kirkwood
Roderick Rozalina

1. Apologies

None.

2. Discussion of Project

One member of the original Coderona team is unable to continue with the project so it will continue with the five remaining members. It is not thought that this will impact the schedule, however, this will be assessed as the project progresses.

API framework options were discussed with the team concluding that further time was required to arrive at a decision about which would be most suitable for the project.

Catriona was elected as the first SCRUM master, tasks were then agreed and allocated to team members to cover the first sprint. These are outlined in point 5.

3. Challenges

Aside from the loss of a team member, no additional challenges were encountered.

4. Team dynamics

Team were happy to be starting the project and the dynamics were positive.

5. Actions for next meeting

All team members are to research API frameworks in order that a decision can be made on this at the next meeting.

Peter will research how the password strength checking may be implemented.

Catriona will research how the password breach checking may be implemented.

Roderick will look at the project schedule in more detail to confirm that the loss of a team member will not impact the schedule.

6. Next meeting

The next meeting will take place on Microsoft Teams at 3pm on the 1st of February 2021.

Coderona Passroclus Project
Team Meeting Week 2 (Online: Microsoft Teams)
Minutes of the meeting held on 1st February 2021

Present:

Stuart Brown
Peter Captain
Tia Cotton
Catriona Kirkwood
Roderick Rozalina

1. Apologies

None.

2. Approval of minutes from previous meeting

The minutes were accepted as an accurate reflection of the meeting.

3. Discussion of Project

API frameworks were discussed after all team members conducted research on the different options available. A decision was made to use the Flask framework.

It was decided that Peter would work on the password strength checking code, Catriona on the breach checking code, Stuart on the web browser extension and password generator, Roderick on the website wireframe, and Tia on the API.

Tia set up a GitHub repository for the team to upload the various project components and an online workspace to record the tasks for each sprint.

4. Challenges

There were no challenges encountered at this meeting.

5. Team dynamics

Team dynamics were good, with progress according to schedule.

6. Actions for next meeting

Stuart will begin work on the web browser extension.

Peter will research password strength checking algorithms.

Tia will begin work on the API.

Catriona will begin work on the password breach checker.

Roderick will begin work on the website.

7. Next meeting

The next meeting will take place on Microsoft Teams at 3pm on the 8th of February 2021.

Coderona Passroclus Project

Team Meeting Week 3 (Online: Microsoft Teams)

Minutes of the meeting held on 8th February 2021

Present:

Stuart Brown
Peter Captain
Tia Cotton
Catriona Kirkwood
Roderick Rozalina

1. Apologies

None.

2. Approval of minutes from previous meeting

The minutes were accepted as an accurate reflection of the meeting.

3. Discussion of Project

Roderick was elected as Scrum master for this sprint.

Peter has written some preliminary pseudocode for the password strength checker.

Catriona has been researching password breach lists and planning how best to implement this given that the files are very large. She has also been researching the best way to implement the string search to ensure that it is fast and reliable.

Roderick has been checking that progress is according to schedule, as yet there are no issues.

Tia has researched and created the first draft of the API.

4. Challenges

Stuart was unable to work effectively this week due to personal circumstances.

Files required to be used in the breach checker were found to be very large, with this creating a potential problem in terms of hosting them on the server. The client advised that smaller files should be used instead, leaving them with the option of upgrading in the future if necessary.

The team is waiting for an account to be provided for the hosting server, this delays the project slightly but should be easily overcome.

5. Team dynamics

Team dynamics were good, with progress according to schedule.

6. Actions for next meeting

Stuart will continue to work on the web browser extension.

Peter will expand on the pseudocode for the password strength checker and implement some of the functions in C++.

Tia will continue working on the API.

Catriona will continue to work on the password breach checker.

Roderick will continue to work on the website.

7. Next meeting

The next meeting will take place on Microsoft Teams at 3pm on the 15th of February 2021.

Coderona Passroclus Project
Team Meeting Week 4 (Online: Microsoft Teams)
Minutes of the meeting held on 15th February 2021

Present:

Stuart Brown
Peter Captain
Tia Cotton
Catriona Kirkwood
Roderick Rozalina

1. Apologies

None.

2. Approval of minutes from previous meeting

The minutes were accepted as an accurate reflection of the meeting.

3. Discussion of Project

Peter has created the first draft of the password strength checker, it currently performs a small variety of checks on a given password.

Stuart has researched and created a simple web browser extension which can be built upon to be used in the final deliverable.

Roderick has been researching and relearning some of the languages necessary for the creation of the website wireframe.

Tia has written the requirements for the client and these should be signed off soon.

4. Challenges

Catriona has not been able to work effectively this week due to personal circumstances but this will be rectified next week.

5. Team dynamics

Team dynamics were good, with progress according to schedule.

6. Actions for next meeting

Stuart will develop the browser add on.

Peter will further develop the password strength checking code.

Tia will continue to work on the API.

Catriona will continue to develop the breach checker.

Roderick will continue to work on the website.

7. Next meeting

The next meeting will take place on Microsoft Teams at 3pm on the 22nd of February 2021.

Coderona Passroclus Project
Team Meeting Week 5 (Online: Microsoft Teams)
Minutes of the meeting held on 22nd February 2021

Present:

Stuart Brown
Peter Captain
Tia Cotton
Catriona Kirkwood
Roderick Rozalina

1. Apologies

None.

2. Approval of minutes from previous meeting

The minutes were accepted as an accurate reflection of the meeting.

3. Discussion of Project

Stuart was elected as Scrum master for this sprint.

Peter has performed additional work on the password strength checker, including implementing common word detection.

Stuart has been working on the browser add on but has encountered an error that has restricted progress.

Catriona has completed a first draft of the breach checker code, it requires some further development.

Roderick has been monitoring the project progress according to the schedule, there are no issues to report on that. Roderick also conducted research on password checkers to plan the design of the website.

Tia has done further work on the API and attempted to install it on the client server, however, the lack of WSGI on the server prevents the API from working.

4. Challenges

A solution will need to be found for the client server issue, the team will check if the client is able to install WSGI.

5. Team dynamics

Team dynamics were good, with progress according to schedule.

6. Actions for next meeting

Stuart will attempt to rectify the error with the browser add on.

Peter will further develop the password strength checking code.

Tia will check if the client is able to install WSGI on the server.

Catriona will continue to develop the breach checker.

Roderick will continue to work on the website.

7. Next meeting

The next meeting will take place on Microsoft Teams at 3pm on the 1st of March 2021.

Coderona Passroclus Project

Team Meeting Week 6 (Online: Microsoft Teams)

Minutes of the meeting held on 1st March 2021

Present:

Stuart Brown
Peter Captain
Tia Cotton
Catriona Kirkwood
Roderick Rozalina

1. Apologies

None.

2. Approval of minutes from previous meeting

The minutes were accepted as an accurate reflection of the meeting.

3. Discussion of Project

Peter has further developed the password strength checker by researching some of the more technical aspects of passwords, such as character pairs and substitutions, and how best to implement that in the code.

Stuart has rectified the error that was identified last week and has created a chrome based browser extension that generates a random string for a password.

Catriona has continued development of the breach checker code.

Roderick continues to work on the website wireframe.

Tia has continued to work on the API which is now hosted on Heroku and connected directly to the API GitHub.

4. Challenges

The API will now be hosted virtually as the original plan to host on the client server was not possible due to the team being unable to install WSGI on that server.

5. Team dynamics

Team dynamics were good, with progress according to schedule.

6. Actions for next meeting

Stuart will develop the password generator to create a more complex password.

Peter will further develop the password strength checking code.

Tia will continue to work on the API.

Catriona will implement handling of multiple text files on the breach checker.

Roderick will create a layout for the website.

7. Next meeting

The next meeting will take place on Microsoft Teams at 3pm on the 8th of March 2021.

Coderona Passroclus Project
Team Meeting Week 7 (Online: Microsoft Teams)
Minutes of the meeting held on 8th March 2021

Present:

Stuart Brown
Peter Captain
Tia Cotton
Catriona Kirkwood

1. Apologies

Roderick Rozalina.

2. Approval of minutes from previous meeting

The minutes were accepted as an accurate reflection of the meeting.

3. Discussion of Project

Catriona was elected as Scrum master for this sprint.

Peter continues to work on the password strength checker and has resolved some errors that were present in the code.

Catriona has implemented the features discussed at the previous meeting into the breach checker.

Tia has finished building the API and can now begin to integrate the code that other team members are working on.

4. Challenges

Stuart has been unable to make any improvements to the password generator this week.

Roderick has been unable to work effectively this week due to personal circumstances.

5. Team dynamics

Team dynamics were good, with progress according to schedule.

6. Actions for next meeting

Stuart will continue to work on the password generator.

Peter will further develop the password strength checking code.

Tia will begin to integrate the breach checker code into the API.

Catriona will complete the breach checker code and perform error checking and performance analysis before sharing it with the team.

Roderick will continue to work on the website.

7. Next meeting

The next meeting will take place on Microsoft Teams at 3pm on the 15th of March 2021.

Coderona Passroclus Project
Team Meeting Week 8 (Online: Microsoft Teams)
Minutes of the meeting held on 15th March 2021

Present:

Stuart Brown
Peter Captain
Tia Cotton
Catriona Kirkwood
Roderick Rozalina

1. Apologies

None.

2. Approval of minutes from previous meeting

The minutes were accepted as an accurate reflection of the meeting.

3. Discussion of Project

Peter developed the strength checker further, it is now close to completion.

Stuart continues to work on the password generator and will convert it from Java to Python.

Catriona has completed the breach checker code and checked it for errors and performance analysis before sharing it with the team. It can now be integrated on the API.

Roderick has created a simple website and has researched the educational content that will be added.

Tia made an attempt to integrate the breach checker code into the API, however, the required library did not work as planned.

4. Challenges

Integration of the breach checker code to the API has proven difficult, other solutions will be explored before the next meeting.

5. Team dynamics

Team dynamics were good, with progress according to schedule.

6. Actions for next meeting

Stuart will continue to work on the password generator.

Peter will complete the password strength checking code.

Tia will continue to work on getting the breach checker integrated with the API.

Catriona will type up the meeting minutes in the required format.

Roderick will continue to work on the website.

7. Next meeting

The next meeting will take place on Microsoft Teams at 3pm on the 22nd of March 2021.

Coderona Passroclus Project

Team Meeting Week 9 (Online: Microsoft Teams)

Minutes of the meeting held on 22nd March 2021

Present:

Stuart Brown
Peter Captain
Tia Cotton
Catriona Kirkwood
Roderick Rozalina

1. Apologies

None.

2. Approval of minutes from previous meeting

The minutes were accepted as an accurate reflection of the meeting.

3. Discussion of Project

Peter will be the Scrum master for this sprint.

Tia has tried to get the breach checking code to work on the API but it has proved problematic due to it being written in C++. She will attempt to convert the code to Python and has implemented a sample breach checker for the time being.

Peter has completed his strength checking code which is written in C++, this will also need to be converted to Python for integration to the API.

Stuart has been working on the password generator which has been written in Python, this is nearly finished and will be shared with the group in the coming days.

Roderick has been working on the website wireframe.

Catriona has been working on the meeting minutes.

4. Challenges

Aside from the integration of C++ code to the API, no other challenges have been encountered this week.

5. Team dynamics

Team dynamics were good, with progress according to schedule.

6. Actions for next meeting

Stuart will complete the password generator and work on the web browser extension.

Peter will work on the web browser extension.

Tia will convert the C++ code to Python and replace the sample code with that, this may involve using a less complex breach checker.

Catriona will finish typing up the meeting minutes.

Roderick will continue to work on the website.

7. Next meeting

The next meeting will take place on Microsoft Teams at 3pm on the 29th of March 2021.

Coderona Passroclus Project
Team Meeting Week 10 (Online: Microsoft Teams)
Minutes of the meeting held on 29th March 2021

Present:

Stuart Brown
Peter Captain
Tia Cotton
Catriona Kirkwood
Roderick Rozalina

1. Apologies

None.

2. Approval of minutes from previous meeting

The minutes were accepted as an accurate reflection of the meeting.

3. Discussion of Project

Peter has been investigating plugins and has created one using Brython, it will require further work to achieve the intended functionality.

Catriona has finished formatting the meeting minutes.

Roderick has been working on the website wireframe and researching ways to properly display the required information.

Tia has successfully integrated a basic breach checker and passphrase creator into the API. She has made progress with the strength checker but it requires some further work before it can also be integrated.

4. Challenges

Stuart was unable to work effectively this week due to personal circumstances.

5. Team dynamics

Team dynamics were good, with progress according to schedule.

6. Actions for next meeting

Stuart will continue to work on the web browser extension.

Peter will continue to work on the web browser extension.

Tia will finish integrating the strength checker code.

Catriona will submit the meeting minutes and start planning the client pitch.

Roderick will continue to work on the website.

7. Next meeting

The next meeting will take place on Microsoft Teams at 3pm on the 16th of April 2021.

Coderona Passroclus Project
Team Meeting During Easter Break (Online: Microsoft Teams)
Minutes of the meeting held on 16th April 2021

Present:

Stuart Brown
Peter Captain
Tia Cotton
Catriona Kirkwood
Roderick Rozalina

1. Apologies

None.

2. Approval of minutes from previous meeting

The minutes were accepted as an accurate reflection of the meeting.

3. Discussion of Project

Peter and Stuart have finished the web browser extension.

Roderick has completed the website.

Tia has finished working on the code integration.

The project is now fully functional and complete.

All team members have been working on the client pitch and will be ready to present to the client on the 20th of April.

4. Challenges

No challenges were encountered.

5. Team dynamics

Team dynamics remain good, the project has been completed successfully and according to schedule.

6. Actions for next meeting

All team members will complete their client pitch contribution.

7. Next meeting

The next meeting will take place on Microsoft Teams at 3pm on the 19th of April 2021.

Coderona Passroclus Project
Team Meeting Week 11 (Online: Microsoft Teams)
Minutes of the meeting held on 19th April 2021

Present:

Stuart Brown
Peter Captain
Tia Cotton
Catriona Kirkwood
Roderick Rozalina

1. Apologies

None.

2. Approval of minutes from previous meeting

The minutes were accepted as an accurate reflection of the meeting.

3. Discussion of Project

During this meeting the team completed their presentation of the Passroclus product for the client, this involved ensuring that the presentation was consistent and rectifying small issues such as sound quality and animation timings. The team were happy with the end result and successfully submitted the presentation to the client.

4. Challenges

No challenges were encountered.

5. Team dynamics

Team dynamics remain good.

6. Actions for next meeting

All team members will complete a first draft of their white paper contribution.

7. Next meeting

The next meeting will take place on Microsoft Teams at 3pm on the 26th of April 2021.

Coderona Passroclus Project
Team Meeting Week 12 (Online: Microsoft Teams)
Minutes of the meeting held on 26th April 2021

Present:

Peter Captain
Catriona Kirkwood
Roderick Rozalina

1. Apologies

Stuart Brown and Tia Cotton.

2. Approval of minutes from previous meeting

The minutes were accepted as an accurate reflection of the meeting.

3. Discussion of Project

Due to personal circumstances Stuart will be unable to proceed with his part of the white paper document.

The team discussed their individual progress with the white paper document and it was agreed that all would complete the work before the next team meeting, with the intention of submitting it to the client that day.

4. Challenges

Stuart is facing a significant personal challenge which means he is unable to work at this time, the team wishes him well.

Tia was unable to attend the meeting but will complete her part of the document before the next meeting.

5. Team dynamics

Team dynamics remain good.

6. Actions for next meeting

All team members will complete their white paper contribution.

7. Next meeting

The next meeting will take place on Microsoft Teams at 3pm on the 30th of April 2021.

Team Meeting White Paper Review (Online: Microsoft Teams)

Minutes of the meeting held on 30th April 2021

Present:

Peter Captain
Tia Cotton
Catriona Kirkwood
Roderick Rozalina

1. Apologies

Stuart Brown.

2. Approval of minutes from previous meeting

The minutes were accepted as an accurate reflection of the meeting.

3. Discussion of Project

The team discussed their white paper contributions thus far and concluded that more time was required to finalise the document before submitting to the client.

4. Challenges

No new challenges were encountered.

5. Team dynamics

Team dynamics remain good.

6. Actions for next meeting

The team will revise the document before the next meeting.

7. Next meeting

The next meeting will take place on Microsoft Teams at 3pm on the 3rd of May 2021.

Coderona Passroclus Project
Team Meeting White Paper Submission (Online: Microsoft Teams)
Minutes of the meeting held on 3rd May 2021

Present:

Peter Captain
Tia Cotton
Catriona Kirkwood
Roderick Rozalina

1. Apologies

Stuart Brown.

2. Approval of minutes from previous meeting

The minutes were accepted as an accurate reflection of the meeting.

3. Discussion of Project

The team discussed the final document and were happy to submit it to the client.

4. Challenges

No new challenges were encountered.

5. Team dynamics

Team dynamics remain good.

6. Actions for next meeting

This was the last team meeting for the Passroclus project, no further action is required as the product has been delivered to the client.

APPENDIX E – C++ CODE

The C++ code that was not able to be implemented is included in the GitHub repository. This can be found at:

<https://github.com/cybersomething/passroclus/blob/main/StrengthChecker/StrengthChecker/StrengthChecker.cpp>

APPENDIX F - RESULTS FROM WEBSITE SECURITY EVALUATION (RODERICK)

Web Accessibility evaluation Tool (WAVE)

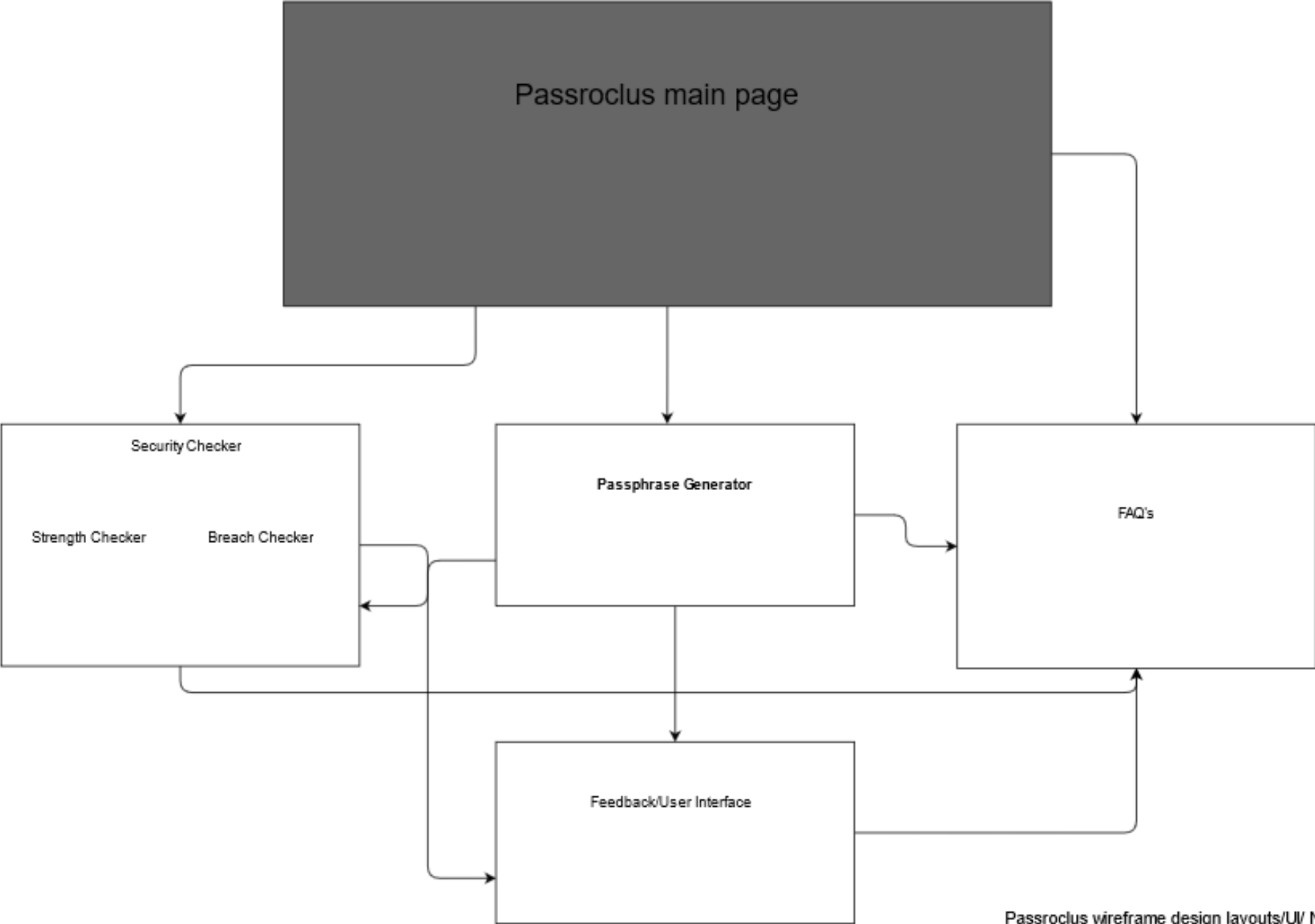
The screenshot displays the WAVE (Web Accessibility Evaluation Tool) interface. The left sidebar shows the tool's logo, the URL `https://passroclus.herokuapp.com`, and a 'Summary' section with the following data:

Category	Count
Errors	1
Contrast Errors	1
Alerts	1
Features	1
Structural Elements	4
ARIA	1

Below the summary is a 'View details' button. The main content area shows the website being evaluated, which has a dark theme. It includes a 'Welcome to Passroclus.' message, a password entry field with a 'Check Password' button, and a section for creating a new passphrase with a 'Create a passphrase!' button. At the bottom, there is a 'Want to read the FAQ's?' section with a 'Go to FAQ's' button and a 'Code' button.

Website Wireframe:

Figure 1-A. Wireframe design from the initial product design phase.



Passroclus wireframe design layouts/UI/ Navigation flow for Browser and Mobile

Final Version; V1

Date completed; 3/5/21

Website Security Check

Figure 2-A

File Edit View Analyse Report Tools Import Online Help

Standard Mode

Sites +

Contexts

Default Context

Sites

https://passroclus.herokuapp.com

GET:/

POST:checker()(password)

POST:creator()

GET:FAQ

generator

GET:2

POST:generatorRedirect()(length)

GET:robots.txt

securityChecker

GET:ZAP

GET:sitemap.xml

Quick Start Request Response +

Header: Text Body: Text

HTTP/1.1 200 OK
Connection: keep-alive
Server: gunicorn/19.9.0
Date: Mon, 03 May 2021 16:21:31 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 1671
Via: 1.1 vegur

<!doctype html>
<html lang="en">
 <head>
 <!-- Required meta tags -->
 <meta charset="utf-8">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <!-- Bootstrap CSS -->
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlo1flfl" crossorigin="anonymous">
 <style>

History Search Alerts ⚙ Output Spider Active Scan +

Alerts (4)

X-Frame-Options Header Not Set (6)

Absence of Anti-CSRF Tokens (5)

Incomplete or No Cache-control and Pragma HTTP Header Set (6)

X-Content-Type-Options Header Missing (6)

WASCID: 15

Source: Passive (10020 - X-Frame-Options Header)

Description:
X-Frame-Options header is not included in the HTTP response to protect against 'Clickjacking' attacks.

Other Info:

Solution:
Most modern Web browsers support the X-Frame-Options HTTP header. Ensure it's set on all web pages returned by your site (if you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. ALLOW-FROM allows specific websites to frame the web page in supported web browsers).

Figure 2-B

The screenshot displays the Burp Suite application interface. The top navigation bar includes 'Quick Start', 'Request', and 'Response' tabs, with 'Response' currently selected. Below this, the 'Header: Text' and 'Body: Text' tabs are visible, with 'Body: Text' selected. The left sidebar shows a tree view of 'Contexts' and 'Sites'. Under 'Sites', the site 'https://passroclus.herokuapp.com' is expanded, showing various endpoints. The 'GET:robots.txt' endpoint is highlighted. The main pane displays the response for this endpoint, showing HTTP status '200 OK' and headers including 'Connection: keep-alive', 'Server: gunicorn/19.9.0', 'Date: Mon, 03 May 2021 16:21:31 GMT', 'Content-Type: text/html; charset=utf-8', 'Content-Length: 1671', and 'Via: 1.1 vegur'. The response body contains HTML code for a welcome page titled 'Passroclus', featuring a form with the action '/checker' and method 'post'. The bottom pane shows the 'Alerts' tab, listing several security alerts. The alert 'Absence of Anti-CSRF Tokens (5)' is selected and expanded, showing details about the vulnerability, including a description of CSRF attacks, other information about the missing tokens, and a solution recommendation to use anti-CSRF packages like OWASP CSRFGuard.

Sites +

Quick Start → Request ← Response +

Header: Text ▾ Body: Text ▾

Contexts

- Default Context
- Sites
 - https://passroclus.herokuapp.com
 - GET:/
 - POST:checker()(password)
 - POST:creator()
 - GET:FAQ
 - generator
 - GET:2
 - POST:generatorRedirect()(length)
 - GET:robots.txt
 - securityChecker
 - GET:ZAP
 - GET:sitemap.xml

HTTP/1.1 200 OK
 Connection: keep-alive
 Server: gunicorn/19.9.0
 Date: Mon, 03 May 2021 16:21:31 GMT
 Content-Type: text/html; charset=utf-8
 Content-Length: 1671
 Via: 1.1 vegur

```

<h1 {color: white;}
</style>
<title>Passroclus</title>
<body>
  <div class="container-fluid" align="center">
    <h1>Welcome to Passroclus.</h1>

    <br>
    <br>
    <form action = "/checker" method = "post">
  
```

History Search Alerts * Output Spider Active Scan +

Alerts (4)

- X-Frame-Options Header Not Set (6)
- Absence of Anti-CSRF Tokens (5)**
- Incomplete or No Cache-control and Pragma HTTP Header Set (6)
- X-Content-Type-Options Header Missing (6)

WASC ID: 9
 Source: Passive (10202 - Absence of Anti-CSRF Tokens)

Description:

No Anti-CSRF tokens were found in a HTML submission form.
 A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF

Other Info:

No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF] was found in the following HTML form: [Form 1: "password"].

Solution:

Phase: Architecture and Design
 Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.
 For example, use anti-CSRF packages such as the OWASP CSRFGuard.

Figure 2-C

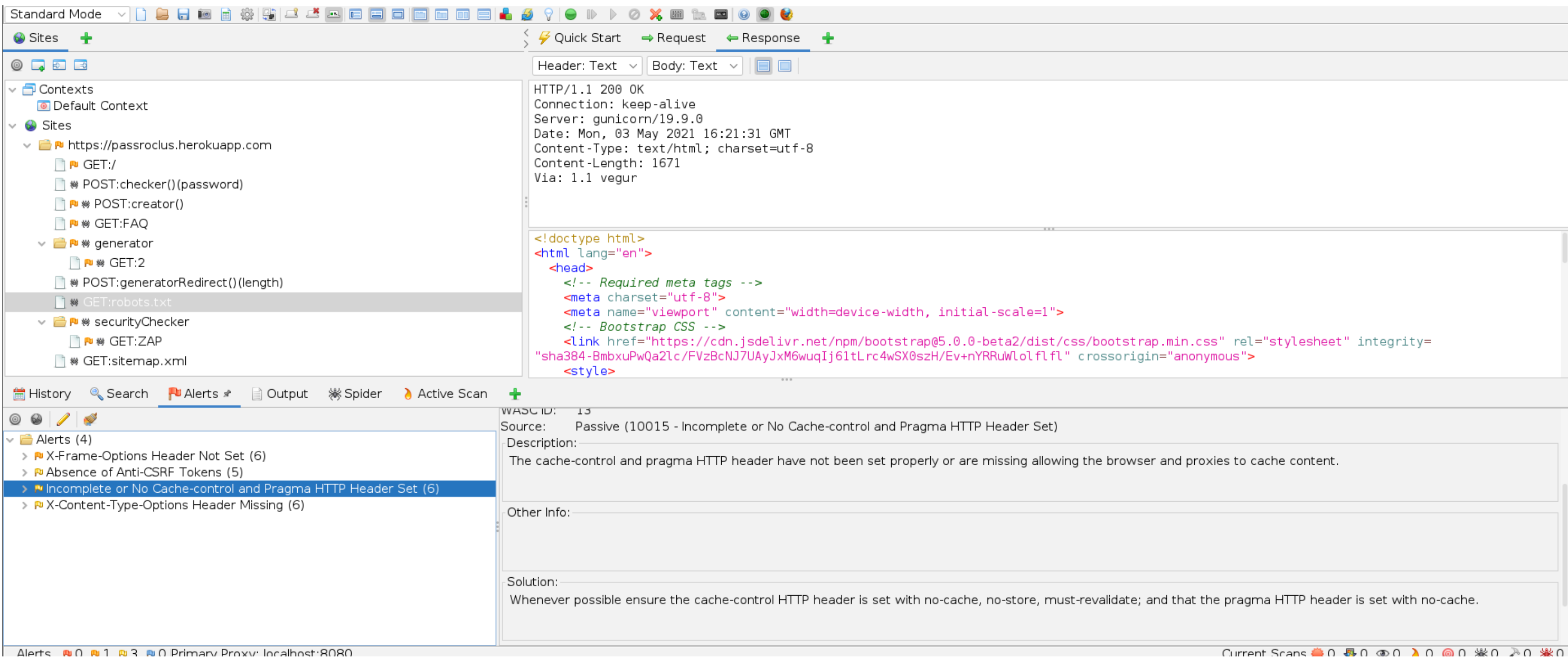


Figure 2-D

