SecuReport!

by

Alexander Vale

Candidate number 2392

Centre number 40531

A Write up of my NEA for
AQA A-LEVEL COMPUTER SCIENCE 7517

in the

Department of Computing
Tytherington School

November 2019

TYTHERINGTON SCHOOL

Abstract

Department of Computing
Tytherington School

A-Level Computer Science 7517

by Alexander Vale
Candidate number 2392
Centre number 40531

This is the write up for my AQA A-Level computer science 7517 Non Exam Assessment (NEA).

It will contain seven sections consisting of:

An Introduction;
A section on analysis;
A section on Design;
A section on Implementation;
A section on Testing;
A section on Evaluation;
A Conclusion;

Within these seven section I will explain my thoughts and findings throughout my NEA project and will document my progress throughout the whole development cycle.

...

Contents

Chapter 1

Introduction

1.1 Introduction

The project I have chosen to undertake is called SecuReport (Secure Report)

The program will be a fully networked client for reporting bullying within a school environment, it will contain a desktop application that is built with C++, combined with a centralized server built using node.js around a mongoDB database which will store messages, logs and resolutions. The database will be fully encrypted and fully compliant with GDPR and Will be accessed from either the C++ desktop client or a JavaScript web client from anywhere over a secure, encrypted connection to the server. The web client (including the centralized server) will be created using a MEAN stack. What this means is that it will use MongoDB as the database (M) , It will also use Express.JS (E), Angular2 (A), And Node.JS (N) (MEAN).

1.1.1 End Users

Potential End Users will be students and teachers within schools. - My main end user will be the Assistant Head, and Head of pastoral care Mr Pilbury. Although the main end user will be Mr Pilbury as the main end user for processing reports, the majority of end users will be students within the school. This is because it will enable victims of bullying, as well as concerned bystanders to report bullying in a

secure and safe manner. As the client is encrypted, it will ensure peace of mind for everyone, as they can be sure that only the specific recipient they desire can open the report. Reporting bullying can be a stressful issue as the bullies often watch the victims to make sure they do not go to a teacher, as a result going to the teacher can worsen the bullying in many situations as it leads to further name calling of things such as snitch.

1.1.1.1 Access To End Users

I will ask students and teachers within the school to use the program and report any bugs or errors, as well as give feedback as part of the critical path of the program which will be vital to ensure ease of use, and continuing development for the program. This will enable all students to feel comfortable with the program, meaning it won't just collect dust.

1.1.2 Programming Skills Used.

Object orientation
Encryption
MongoDB
Databases
Central database
Sockets
Networking
Security
Complex algorithms (encryption)
Complex client server model
Complex Mathematical processes (Asymmetric + Symmetric)
Sorting and Searching
Recursion
hashing
Web Server Configuration

1.1.3 Technologies used

Symmetric Encryption
Asymmetric Encryption
SSL
Cassandra
Windows Server
MEAN Stack (MongoDB, ExpressJS, AngularJS, NodeJS)

1.2 Introduction to other Sections

Analysis
Design
Implementation
Testing
Evaluation
Conclusion

1.2.1 Analysis

In this section I will analyse current systems in use, including those at my current school, and other schools in the local area.

1.2.2 Design

In this section I will Explain my thoughts throughout the design process, as well as provide user feedback.
I will also explain What processes and technologies I will use to complete my programming assessment. I will use psuedocode as well as Flow charts to explain this.

1.2.3 Implementation

This section will explain any programming I have done, as well as screen-shots of my programming working.

1.2.4 Testing

In this section I will include all test plans for my software, as well as my carrying out these tests, and will include screen-shots of each test.

1.2.5 Evaluation

In this section I will evaluate My Program, and suggest ways to alter my program to make it more secure, or more functional, based on user feedback and testing.

1.2.6 Conclusion

In this section I will evaluate the whole process of creating the Program and suggest things I would do differently the next time around.

Chapter 2

Analysis

2.1 Research methods

One research method that I will use will be Questionnaires, this will help me to establish how students feel about reporting bullying. They will help me to establish this through concise meaningful questions such as do you feel comfortable about reporting bullying. These questionnaires will also help me to develop an understanding of what my system will need to include, such that the program is usable for every student regardless of their ability with computers, this will be made possible through questions such as "please suggest some improvements to the current system".

Another research method I will use will be Interviews of both students and teachers. These will focus on how bullying is reported, including their opinions on the current system, such as; what they like about the current system and how it could be improved. I will conduct these interviews in both a one to one environment, as well a one to many environment, as this may make people feel more comfortable to discuss the issues regarding the current system of reporting bullying. To help ensure true unhindered opinions are given I will not use any names in my report, just initials, and remove any personal details and change names in any examples given my students.
I will also research HCI to establish what I need to do, to ensure that my system is user friendly for school children, who may not have a deep knowledge of

computers. One Such thing that will need consideration is the current operating system and software within school, as most students are probably happy to use these packages and will know instinctively how they work. The final research method that I will use will be Secondary Research - this will include reports from anti bullying charities, such as Ditch the label, as well as from the government in relation to education and their advice and guidelines regarding bullying.

2.2 points to do/research

- data flow diagram
- data sources and processes
- data dictionary

2.3 Questionnaire

See Appendix .1 for the screen-shots of my questionnaire, including the answers I received and some lovely pi charts.

2.4 Interviews

See the Appendix .2 for the interviews that I will give to Teachers and students to collect data about the current system for bullying.

2.4.1 Teachers

See the Appendix .2.1 for the Teacher Interviews

2.4.2 Students

See the Appendix .2.2 for the Student Interviews

2.5 End Users

My End users will be split into two categories - Teachers, which will be the Primary End Users, and Students, which will be the Secondary End Users, by inheritance parents of the students which may also use the Program will also be Secondary End Users.

See Appendix .3 for an email correspondence with my primary end user, confirming that they are happy to be the end user. :

2.5.1 An Appreciation of the skills possessed by my end users

Within the school, all computers run either windows seven complete with the Microsoft office 2010 software suite, or windows ten complete with the Microsoft Office 365 (2016). The Computers all use either the Edge browser provided by Microsoft, or the Google Chrome Browser as downloaded by the technicians from Google. As such the web application will be optimised for these browsers to ensure perfect use within school. As the main software package used is Microsoft Office, the C++ application will be visually optimised to look like one of these, as far as HCI (see section 2.11 Reseach about HCI and UI design)

2.5.2 An Appreciation of the Requirements demanded by my end users

In order for the Program to be used to its maximum effects, the program must be created to a specific set of guidelines as governed by my end users. As such, I sent a research survey out to a selection of my end users. The following are their demands and subsequently the requirements of the program;

- A Web Application as well as a Desktop Application, however as my primary end user has requested a Desktop Application, this will only be a secondary focus and will lack full desktop functionality.

- An option to either submit reports anonymously, or to log in and then receive updates on the report process.
- An option to choose which teacher they can submit the report to.

2.5.2.1 Limitations of the demanded requirements

The limitations of the program, and features that will not be available within the program (as specified within the critical path - they may be part of the non-critical path that will be implemented later) will be as follows:

- Full Web-App Functionality
- The system will not preserve logins, as the computers within the school are not individually assigned and anyone can use them, so it would be immoral and not best practice if a user could see someone else's reports.
- The Program will not rely on LDAP or Active directory to manage user accounts, as these will be completely different for each school and as such there will not be a nice way to have it automatically configure. Students also may install the App from home, where the computers do not sit within active directory, so would have no way to log in.
- There will not Be a Phone App as mobile Phones are not permitted within the School. This rule was Implemented as part of the schools crack down bullying - specifically cyber-bullying. As phones are not permitted in school, it will be a waste of development time to create such an App, when it could be used on far more important things.

2.5.3 Use Cases

The following information will be the use cases of the program, highlighting exactly what each type of user will be able to do. This will be useful for the implementation stage of the development cycle as they

2.5.3.1 Teachers

2.5.3.2 Students

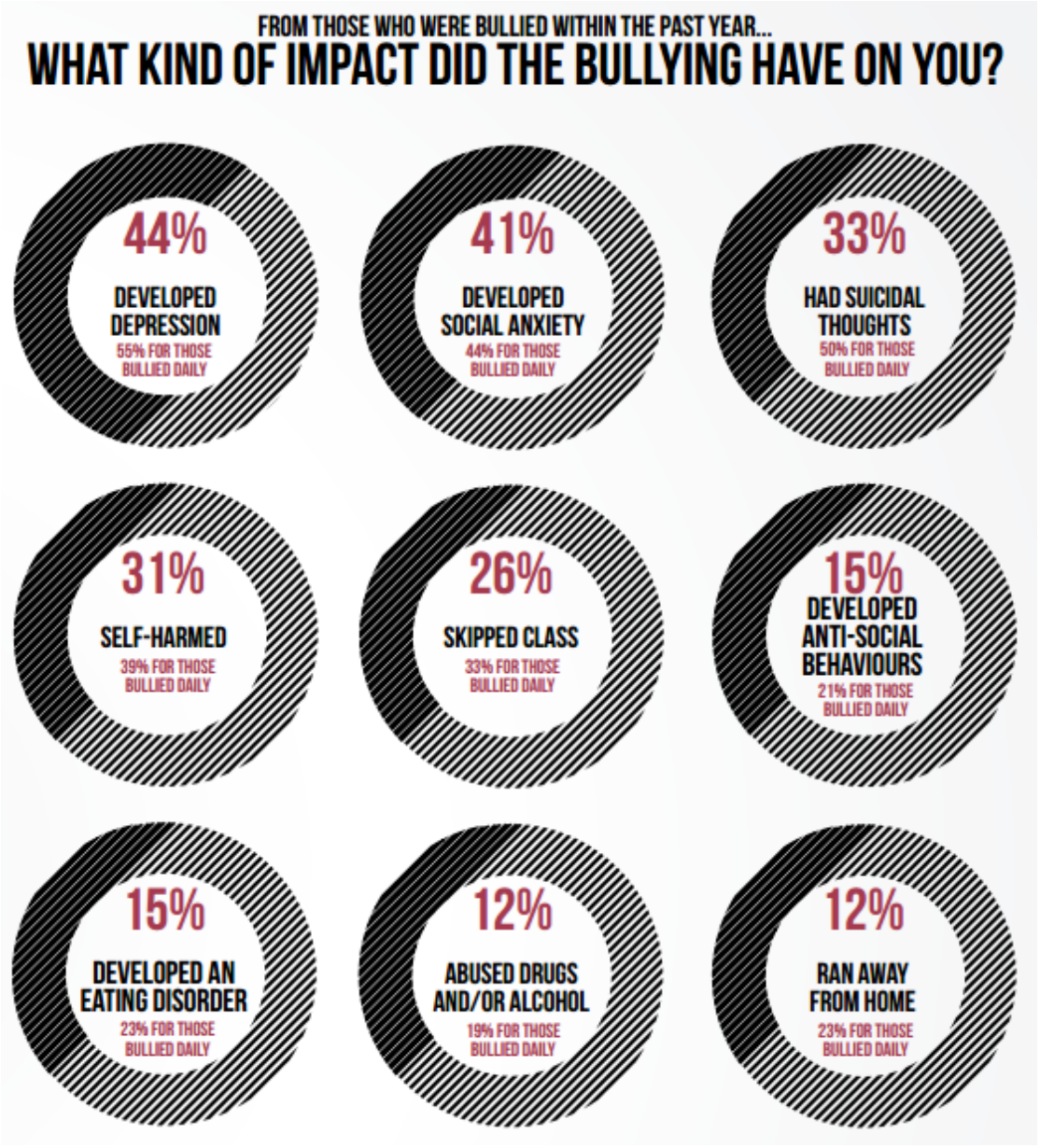
The Students Requirements are to firstly to choose whether to log in. Then they must be able to submit a report, receiving updates and further support such as requests for further information. They alternatively should receive an option to

2.6 Bullying within schools

Here are some statistics which highlight the devastating effects of bullying and shows just how widespread the effects are.

- 1.5 million young people (50%) have been bullied within the past year.
- 145,800 (19%) of these were bullied EVERY DAY.
- People who have been bullied are almost twice as likely to bully others
- Twice as many boys as girls bully (66% of males vs. 31% females).
- 57% of female respondents have been bullied, 44% of male respondents and 59% of respondents who identified as trans have been bullied.
- 24% of those who have been bullied go on to bully.
- Based on their own definition 14% of young people admit to bullying somebody, 12% say they bully people daily.
- 20% of all young people have physically attacked somebody.
- 44% of young people who have been bullied experience depression.
- 41% of young people who have been bullied experience social anxiety.
- 33% of those being bullied have suicidal thoughts.

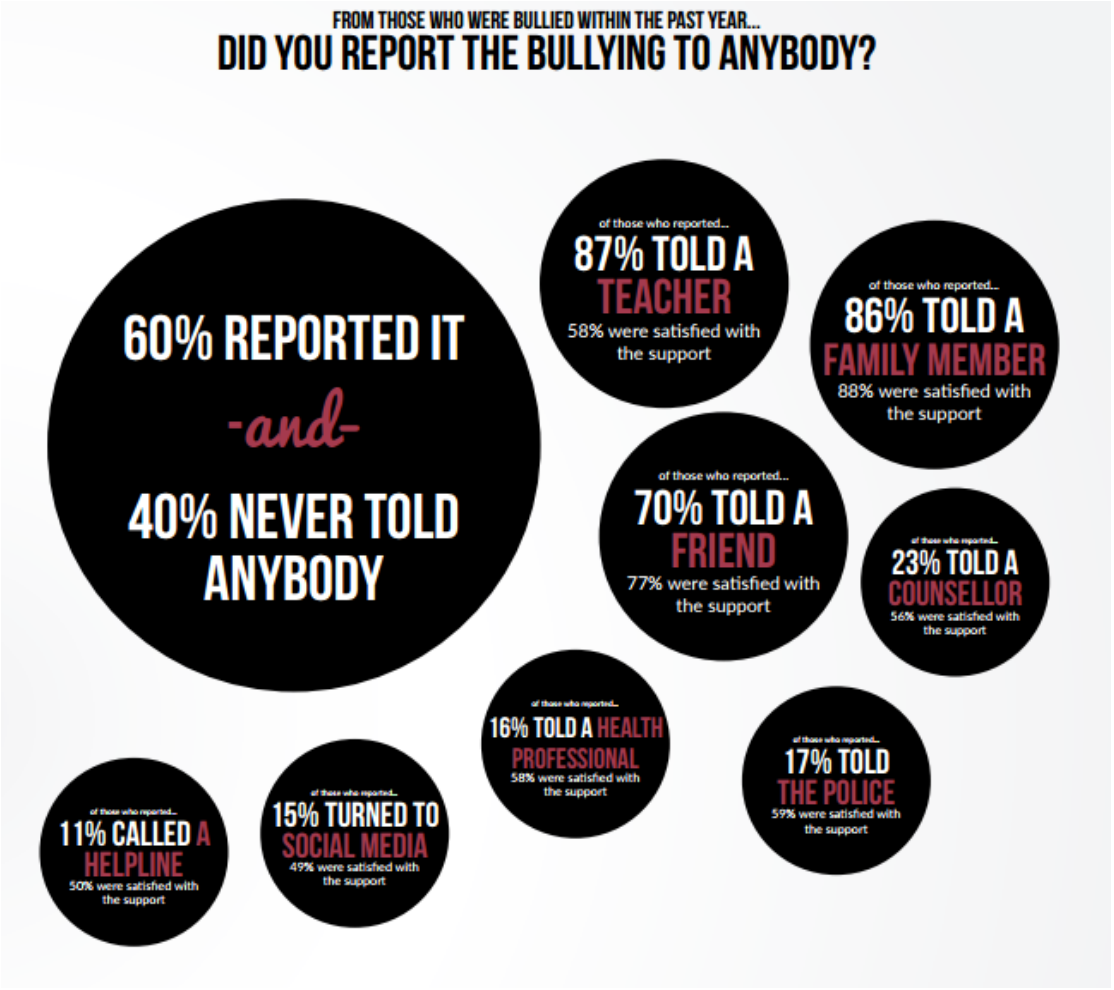
statistics obtained from Ditch the label



The following data has been obtained from the 2016 report by Ditch the label.This data supports the research I did with my own surveys, but is important for me to include as it shows a much larger data set.

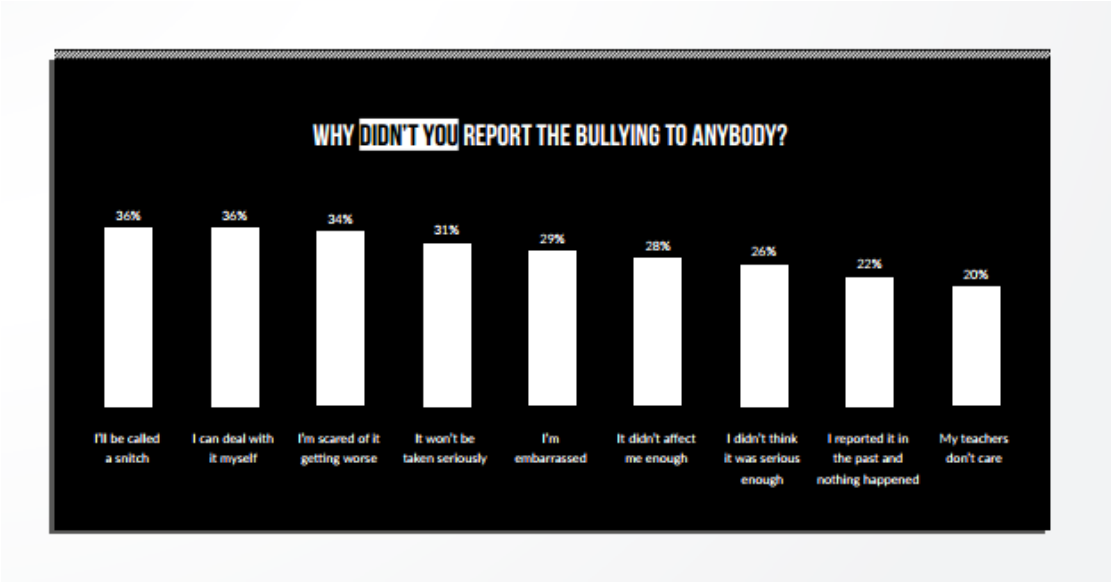
This info-graphic shows the effect of bulling on people and shows the devastating effect of bullying. This motivates me to make my program the best it can be, so that I can help my end user eradicate bullying within the schools they work.

This info-graphic shows the number of people who reported bulling and whether or not they were satisfied with the support they received from the person they reported the issue to. The highest percentage of being satisfied were if they student reported the issue to a family member. However as this cannot be achieved by my



program, and as the nature of family means they student can report this within the safe confines of their home, I will ignore this statistic. The graphic also shows that 40% of people never told anybody, which in my eyes, given the nature of bullying and its effects, highlighted in the info-graphic above, needs to be changed.

This info-graphic shows the number of people who did not report bulling as well as the reasons they did not. The biggest reason at 36% was that they were scared of being called a "snitch". This is a problem that would be avoided through the use of SecuReport as the "bully" would not know that they have been reported and has no way of finding out, so they school could say that they "observed the bully" whilst they were investigating it. Another reason people did not report the issue to anyone is because people are "scared of it getting worse" this is also an issue that could be avoided with the use of SecuReport. The reason behind this is that the nature of the program allows the user to report the issue in a very detailed and efficient manner and they know with confidence that the issue will be dealt by someone will safeguarding training and that it will be done in an utmost



professional manner. Another reason people did not report them being bullied was because they believed that the "teachers don't care" this is a misconception regarding the teachers and simply by the school adapting their policy towards the use of SecuReport shows that the teachers do in fact care. SecuReport also makes sure the issue is escalated to someone within the school who has a safeguarding orientated job, meaning that they REALLY DO care about the students as they specifically chose a job which allows them to help and protect young people.

2.7 The Current System within Tytherington School

Where an incident of serious negative conduct occurs, the following procedure is to be followed:

1. Parents to be informed immediately that an incident has occurred, is being investigated and that they will be contacted again once the investigation is complete.
2. Where appropriate, students involved in the incident to be situated in the Isolation Room in order to facilitate;
3. A full investigation led by an appropriate member of specialist staff or a middle/senior leader.

4. A nominated decision maker (i.e A Head of Year or Senior Leader) to consider evidence from the investigation and decide upon outcomes or refer up where appropriate.

5. Students and parents to be informed of the outcomes that are a result of the serious negative conduct and a record to be made within SIMS.

6. Support of students in a return to positive conduct and their normal educational experience which can include elements of alternative provision.

In the event of any period of alternative provision or exclusion from a students normal educational experience a full and complete record is to be kept in order to allow for analysis against the following objectives:

- To support a return to positive conduct with key indicators demonstrating improvement in conduct.
- To continue to provide a learning experience that allows academic progress to be made

Taken from the Tytherington School document on

Anti-Bullying Policy & Procedure

2.7.1 What this means for my program

While reports can be reviewed by any member of staff, they must be escalated to a senior member of staff or to an allocated decision maker, because of this Evidence must be uploaded to be reviewed. As the current system notifies parents of the outcome, my program must be able to send notification to parents as well, unless it is found to be a better solution for the decision maker to contact the parents directly through, for example but no limited to, a phone call. This would ensure that the decision maker can explain fully, and ensure the understanding and cooperation of the parent, whereas an email may go unnoticed in an Inbox for a few days, which could lead to them violating an exclusion for example. This will need to be researched further and discussed with my end users, however if an emphasis is put on direct conversation between the school and the offending childs

Table 2.1: Data sources and Destinations

Piece of information	Source	Destination
User Details	User enters details into program	sent to server to validate
User login authentication	Sent from server	User that logged in with valid credentials
Initial report	user submits report	Stored in db, sent to 'pool' of reports
Automated reply	Automatically created by server	Sent To User
Teachers report	Uploaded by teacher	Sent To decision maker
Final Report Decision	Decision makers Reply	Sent To Teachers involved in dealing with the report Sent To Pastoral Head
Reply To Student	Decision maker / pastoral head	Sent to the Student that initially submitted report

parents, then issue can be resolved much more effectively and efficiently, as the parent can then discuss the issue with the child, and the child may respond better to a parent, rather than a teacher that they might feel retracted from.

2.8 Data

Given the previous section regarding the fair use and the new data protection regulation coming into law imminently in the EU, all the data I will collect will be lawful and will be subject to the regulation specified by GDPR.

2.8.1 Data Volume

At the school there are 1200 students, and each of these will need to have a login and profile. This will take up relatively little space compared to all the reports, and decided actions, however most students will not create reports as the school is a safe environment and for the most part is relatively bully free. Each student will have an email, as well as a password and a year group.

2.9 Encryption

There are many cryptographically secure algorithms that are industry approved, such as RSA and AES, to distinguish between them, I have written the following section to explain them:

2.9.1 types of encryption

There are two main types of encryption algorithms, symmetric and A-symmetric. Symmetric uses a single encryption / decryption key which must be physically transferred by the two parties in order for encryption / decryption to be used. A-symmetric uses two keys, one public key and one private key, which allows for anyone to hold the public key and they cryptography to still be secure.

2.9.2 my implementation

I will use a symmetric approach as it will be easier to ensure that it is cryptographically secure. Whilst this means that everything depends on the key being secure, I will ensure that this Is overcome, by the the server sending half of the key to the client, which will not be able to be intercepted. This will ensure that if anyone is watching the client generate a key, they will only have half the key. To find the next part of the key I will perform a complex one way mathematical operation on both half of the keys, forming them into one key.

2.9.3 Initial thoughts regarding encryption system

create base 64 of message and a base 64 encryption key. Raise the encryption key to the power of the base 64 of the message, giving the encrypted message in base64 (can be left in base 64 for storage, otherwise can be converted to UTF-8). To get the plaintext back, the encrypted message would be logged to the base of the encryption key.to add extra security, I will perform an exclusive or on the message with a padded (meaning the message can be any length) base64 string that is kept private to ensure secure cryptography.

To ensure the security of the encryption keys, these keys would be encrypted again, using a different algorithm, and then stored in a local database, which only the server can read.

Here is one possible route I may go down for my encryption:

Encryption key$^{plain-textmessage}$ = encrypted message in base64

$\text{Log}_{encryptionkey}$ Encrypted message = plain-text message

2.10 Password storage and hash functions

Storage of passwords is an ever evolving process as industry standards change.

2.10.1 Plaintext

From the early ages of the internet, passwords were stored in plaintext, which nowadays is considered an atrocity. This is a terrible idea as if there is a database breach, then peoples passwords are easily accessible. This is particularly problematic given the nature of humans and passwords. What I mean by this is that most people will re-use one password for everything, and some of the more technically inclined people will use different passwords for what they consider to be important and not important sites. For example, they might consider their online bank to be important, so would use a password for only that, whereas they would use the same password for all their social media accounts. This is terrible practice as if there is a database leak and the same password is used elsewhere, then fraudulent activities may occur with their accounts.

2.10.2 encrypted

Following the era of plaintext passwords, came a period where passwords were encrypted, however this also has its limitations. One such limitation is that if all the passwords have the same encryption key, then it may be possible to figure that

out. For example, if 50 people have the same password that cannot be read as it is encrypted, but they all have different password hints, for example genre of movie, Michael Jackson song then it may be possible to guess the password is thriller then it would be possible to reverse engineer the encryption algorithm and then decrypt all the passwords in the database. Following this happening the scenario explained above could happen.

2.10.3 hashed

Recently encrypting passwords has been back seated in favour of salting and hashing. This is where a complex one way mathematical function is applied to the password and some unique, yet reproducible data is added. This means that if the database if breached one or two passwords may be guessed by hints, but the algorithm cannot be reverse engineered.

2.10.4 my Implementation

To keep in line with industry standard I will salt and hash all passwords using my own hashing algorithm. One potential salt I will use is the first 500 digits of tau (because pi is too mainstream)

2.11 Research about HCI and UI design

In this section I will explain research I have conducted into Human-computer interaction and the design principles of graphical user interfaces. This research is important as it will enable my program to be extremely usable, as people will be able to 'pick up' the program and use it effectively from the start as it will be intuitive from the start and will follow many universal principles.

the main Design principles are as follows:

- way finding
- feedback

- visibility
- consistency
- mental model
- proximity
- grouping
- mapping
- affordance
- progressive disclosure
- 80/20 rule
- symmetry

2.11.1 HCI

the human computer interaction process can be either a very stressful or very pleasant experience. Every human develops a mental model of everything they use. This means that they expect certain things in programs they use such as menu's on the left. As a result of this, we should not vary from the mental model of the masses. What this means is that a person who uses apps everyday has a definitive model inn their mind of how an app should work. It the app fits this model it is a good app. If it doesn't fit this model it is bad, and un intuitive, even if it has extra functionality and WOULD be better if the user took the time to learn the new model. Due to human nature, the user would not learn the new model, they would instead cast it aside for something that fits the old model. 80/20 rule 80% of the usage of the app is created by 20% of the functions of the app. The 80% that doesn't give much usage can be hidden.

in order for it to be a pleasant experience following the mental model of the masses, my app should offer:

- a way-finding system
- feedback

- consistency - fonts, glyphs
- symmetry

2.11.2 UI Design

Apps should be intuitive, and provide clear, easy access to all necessary information. There should be an ever narrowing focus of information, following a way finding system, with easy access to the main page and the exit if they wish. The system should become more specific the deeper into the program they go.

The main principles of a way finding system should be:

- what page am I on now
- what pages can I go to from this page
- what will be on that page
- how do I return to the previous page
- how do I go to the main page

(page refers to a web page or a view of the desktop client)

in order to be intuitive and easy to follow I should follow industry standards and practises such as:

- use descriptive labels
- offer options to go 'back' a page
- offer options to return to the main menu of the app
- if there is a menu system this should be on the LEFT hand side.

the app should give feedback, which allows the user to anticipate problems. Feedback should be timely and specific. It also MUST be VISIBLE. Some valid feedback would be:

- status
- completion
- warning
- errors

2.12 Current Software UI Case Study

Within this section I will discuss software that is currently used within the school here at Tytherington, which students will also be familiar with. I will also discuss some software I frequently use outside of school as I believe this can often have a greater impact to productivity and ease of use. I will include Screenshots of each application within the appendices. Note: All Screenshots taken on windows 10, and all contents belong to their respective copyright holders, notable:

Adobe - PhotoShop
Google - Google Chrome and Google.co.uk
Microsoft - Windows 10 and Associated Files Microsoft Office Productivity suite
Discord - DiscordApp
Atom - Atom Editor

2.12.1 Software Used Within School

The Main software running within Tytherington School is the Office Productivity suite (Office 365 - 2016 version) running on the Windows 10 operating system. As a result of this, the software that I create MUST work on windows 10, and have a Microsoft office feel to it. Other software installed on the school computers that students may well be familiar with may be Google Chrome or the windows 10 File Explorer.

Appendix 1 . File explorer is the bundled utility for navigating through files within windows 10. This means that if the student has to upload a file to my system, containing evidence for their report for instance, then the interface they do this through should look similar to File explorer to ensure ease of use for all students.

Appendix 2 . Google Chrome is a web-browser created by Google, inc. The technicians have opted for this web browser compared to the bundled browsers offered by Microsoft - edge and internet explorer, as they both lack functionality. They are also not very easy to use, as a result of their cluttered user interface, with many buttons that are not needed and rarely used. Whilst my application will not share any functionality or features with Google chrome and will end up having very different interfaces, I believe that it is a very good example to include as it perfectly demonstrates clean and clutter free UI design, which I aspire for my product to have

Appendix 3 . This Appendix is Microsoft office Word, showing a blank page and its very clean, neat taskbar at the top. This shows that while the user is focused on the document in the middle of the document, their view is not disturbed and they get a clean view of what they are working on. This should be a feature of my program, for instance when the student is writing the report, this should be done so using a clean user interface.

Appendix 4 . This Is the Microsoft office new page. It shows a list of recently edited pages, and gives some templates for new files. This will not be needed in my application as all reports will be constructed using a strict template of different text boxes, similar to a web form.Despite this I feel it is important to include as it is another very good example of good good interface design, with good spacing between features that may be clicked on.

2.12.2 Software Used Personally Outside of School

Appendix 5 shows a screenshot of my favourite editor, Atom. There are many reasons for this, including the clean interface, and customisable styles. One option that makes it my favourite is the option for a dark theme. I find this makes the program much easier to use for long periods of time and reduces the risk of eye strain. Whilst the application I am creating will not be used for extensive periods of time, it is an option still worth considering as it can help some people that are partially colour blind spot differences in shades. However despite this, some people may find a dark theme harder to look at, so if it is implemented, it should be an option that can be toggled within the application. Given the nature of this, it falls outside of the critical path and I believe it is a ”nice to have” feature, and should be dealt with accordingly.

Appendix 6 shows an application called Discord, which also has a dark theme. Discord is primarily a text and video platform for online PC gaming. Regardless of this it has a very nice UI, and also has some very informative blog posts. As both applications handle text, I believe them to be comparable. This particular screenshot shows the settings page, in particular settings regarding the user interface of discord. One toggle-able option is the option of a dark theme or a light theme. This will be how I will implement my theme selection if I decide to create both a light and dark UI theme. Appendix 7 shows the main page of discord. Some personal information has been removed and replaced with text explaining the feature that were there. For example the list of servers that I was a part of has been removed, and has been replaced with the text "list of servers". Despite this I still believe the screenshot demonstrates good UI design as all the major features are clearly accessible and are easy to understand. There is also a large area in the middle of the screen for past messages to be viewed. This is one feature that I like so much, that I might take inspiration from it when creating my GUI.

Appendix 8 shows a program called Adobe PhotoShop CC. PhotoShop features a dark theme by default and has a very clean interface, and demonstrates that interfaces can be fairly empty and still be beautiful. There are also two tabs, one named "work" and one called "Learn". When clicked on learn, the application displays a very intuitive and methodical walk through of the program, which is something I will consider doing for my application, as there will be students that may not be familiar with Desktop Applications, as they mostly use touch-based input devices such as tablets and smart phones. Appendix 9 also shows PhotoShop, but rather than the blank home screen, it is showing the new page, and it is a very clean intuitive design. Whilst many of the options on the screen would be extremely irrelevant, as they relate to image dimension and options, I still believe it gives a clear view of clean modern UI design, and the use of a second overlay to create a new document appeals to me so much, that I will take inspiration from it and have an overlay for my new reports as well.

To conclude, when creating my GUI I should ensure that it is clean and clutter free, possibly containing an option for a dark theme. This will ensure that the program is easy to use for all users, including those that are potentially new to desktop PCs and have little experience with desktop applications on windows. I will also include a GUI tutorial the first time someone signs in, with an option within the page to open to an HTML help page, that is either bundled with the

software for offline use, or hosted online, for continued development evaluation, and support.

2.13 GDPR

General Data Protection Regulation (GDPR) was passed into British law in 2016. Despite currently being British law, it will only start to be enforced as of may 2018. It substantially increases the protection for the individual compared to the data protection act, as it is individual privacy centric. Large companies such as Google and Facebook will have to fundamentally change the way they operate in Europe as a result of the GDPR legislation. Brexit will not affect GDRP as it is already British law, meaning that for my project I will still have to make sure I am GDPR compliant. However, whilst this could change in time, as Britain is likely to want to continue to trade with Europe, Britain is likely to want to keep in step with European privacy law.

2.13.1 Compliance

To be GDPR compliant I must:

Explain to users what information I am collecting about them Explain to users why I am doing so and what I will use the data for. Any data I hold about users must be accurate. To not use users' information if they opt out and do not want to. If a user opts to have their information deleted, I must do so. As I will not be charging for this service, I will instantly drop their data. I must have explicit consent that I can use their information for any purpose. I can deny access to the system if they do not give permission, so long as I am reasonable and do not exclude people unreasonably.

2.13.2 rights protected by GDPR

The GDPR particularly protects the rights of children and ensures they have the right to be forgotten.

CIA Confidentiality, integrity, availability

As my system will be for reporting bullying. There will be a definite definition of who is classified as the victim and who is the perpetrator. Therefore It becomes complicated as people may make false reports to tarnish someone elses reputation. So the question becomes whether or not it is justified to keep logs of reports, to show whether this is done for legitimate purposes. To ensure I am not bias and completely fair, I must adopt the attitude of innocent until proven guilty for my system and software. As a result I need to find a balance between freedom of information and data privacy.

Given the sensitive nature of the program, how will I not profile people?

For my system and my purposes I will treat anyone under 18 as a child, which is the highest threshold in the E.C. To collect any piece of data I should be able to justify it and give strong argument in favour.

To make sure I am compliant with GDPR I should carry out a privacy risk assessment, which means assessing what data I am / will be holding and question whether I need to hold it. If I dont, get rid of it. If I do need it, can I justify why I am holding it. I should also ensure that whoever moderates the reports for the school has enough knowledge to judge each report without bias and in a neutral manner.

For not being compliant with GDPR I risk significant fines which are unprecedented levels. Of which ever is higher out of 4% of annual turnover or 20 million euros.

If the organization has more than 250 people a data protection officer must be appointed. The data protection office is a formal officer, who must respond to ALL GDPR inquiries. They are accountable to the board and to the regulator (ICO within UK). As SecuReport.space is an organisation with less than 250 employees, I will not have to appoint one. However, I should maintain a record of what data I keep and what it is used for.

I should also have a declaration when a user signs up that they must agree to saying that each report must be submitted with utmost truth and no lies should be told to tarnish someone elses reputation. And also, I must adopt a policy and procedure which defines and governs the reporting. This should also reflect how any reports submitted will go through to a teacher, whether they are factious or not and regardless of content. At the teacher level within the school, with safeguarding

officers is when issues will be judged and dismissed if they are deemed to be invalid or false reports.

2.14 discussion of languages

2.14.1 Desktop

within this section all languages will be assumed to be created specifically with windows in mind. Cross Platform usage is a bonus that would be nice to have, but is in no way a part of the critical path for the program.

2.14.1.1 C++

C++ is the lowest level language that I am considering using. It is known for its insane speed at run-time, though must be compiled. C++ allows for me to use Object orientated programming, which will be a vital asset in my development cycle.

2.14.1.2 python

Python is the highest level language on this list. It is fairly slow during run time, due to the fact that it is interpreted rather than compiled. Despite this I am most familiar with python and as such will require me to learn less features, comparatively to the other languages discussed. In addition to this, being a high level language it is very quick to write and easy for anyone else to understand the code. Python would also allow me to use object orientation, however pythons implementation of OOP means that the

2.14.1.3 JavaScript - Electron Framework

Using electron would enable me to code the application once using JavaScript and use that for both the desktop and the web-client. JavaScript can be used to program in an object orientated manner. This will enable me to create an

object for a report, then use this as a template for users filling in a report, then submitting that into a database (notably mongoDB)

2.14.2 Web

Within this section all given languages will be build around HTML5 and CSS3.

2.14.2.1 JavaScript

2.14.2.2 PHP

2.14.2.3 Python - Django Framework

2.15 Potential Solutions

2.15.1 Web Application Created With PHP and MySQL (LAMP server)

In order to solve the problem of all students having different platforms and devices at home, there is merit in the idea of creating a uniform web Application, that is usably from any device with only a web browser. One downside to this is that my end user has specifically requested a Desktop Application so that would have to be created as well.

2.16 Final Solution

With regards to my end user, I will create a desktop application with C++ aside a lighter web client that will be built using JavaScript. This will give every user the ability to report bullying from any device, as well as A application that behaves similarly to Microsoft office. This is due to items described in section 2.5.1. The Solution will rely on a mongoDB database as this has shown high performance in other non relational programs such as online chat applications. For example discord in its early days, before they switched to using Cassandra. While they Stated

Many Reasons for switching, their main reason was that latency was increasing and becoming unpredictable, Though this was only apparent once they hit around 100 million messages saved. This is highly unlikely to happen within the scope of my project. As a result of this the downside of MongoDB, of poor scalability through sharding, can be ignored.

2.17 Critical Path

- Set up server with web page (dynamic for each school, on a sub domain)
- JS and html functionality of client.
- Create basic functionality for client and server, where client can send message to server
- which displays the message and sends a confirmation, but does nothing with the message
- Create two databases:
- one as a login server
- one for message storage
- Create message storage functionality into server.
- Create ability to push messages from the server to the client when requested.
- Create anonymous online submission form.
- From client users can access message logs, view past reports, including details and what the end result was.
- From web client, users can only submit anonymous forms, and cannot view them once sent.

2.17.1 Non-Critical Path

- ENCRYPT EVERYTHING
- Extend web client to full functionality.
- Email functionality

2.18 SMART TARGETS - Specific, Measurable, Attainable, Relevant and Timely

- Further note - after the completion of objectives 4 5 6 and 7 , there will be beta releases of the program to receive feedback, after which some objectives may be revisited to add extra functionality which is desired by end users.

2.18.1 Objective One - Set up Server

My first SMART objective will be to set up both the server for the database and the web page / client. As the server will also be a scripting language to receive reports from desktop clients, it must be running a suitable operating system, which can run the said scripting language as well as a web server. So, the decision of which operating system as well as any additional software shall be encompassed inside this objective.

This objective will be measured on whether the server is setup and has all software installed. Once all software has been chosen and installed onto the server this objective will have been attained. This will be checked by running a test web page that passes information to a scripting language, which then returns it to the database and stores it.

This objective is relevant as without the server none of the functionality of the program will be able to work. The objective should be completed within 2 days from the beginning of the implementation stage.

2.18.2 Objective Two - Set up Databases

My second SMART objective will be to decide on which database I will use, out of SQL and NoSQL, and further which implementation I will use. Once I have decided which Database to use I will install the software onto the server and then setup the databases. I will use two of them as one is needed to enable login functionality- storing emails and passwords, the other will store messages and server logs.

This objective will be measured on if the databases allow a user to login and store data. When a user can do so, this objective will have been attained.

This objective is relevant as without a database, no one will be able to login and reports will not be able to be saved. This objective should be completed within a day of the first SMART objective being completed.

2.18.3 Objective Three - Server Side Script

My third SMART objective will be to create a script using a scripting language on the server that receives connections from the client. Depending upon the clients request, if the client is sending a new report the client stores this in a database (and possibly emails them), if the client is requesting new reports then the server will send the client an update on either their report status, or sends them a reply if they are a student, or pushes and reports submitted if they are a teacher with report read access (as defined by the head of safeguarding at the school).

This objective will be measures on if the server pushes reports to clients or stores them in the database. *As the client will not have been created at this stage, a small script will be created that sends test data to the server, as well as sends a pull request. This objective will be attained when the scripts test data passes.

This objective is relevant as without the server being able to send and receive client queries, then the program will not be able to work, as reports will have no way of reaching a teacher. it would be like trying to send a message via carrier pigeon without a pigeon. This objective should be completed within 5 days of the completion of Objective Two

2.18.4 Objective Four Client Back end

My fourth SMART objective will be to create the client side script / back end. This will be what the user sees however at this stage will be headless and will not contain any GUI as this is not *crucial* to program functionality. The functionality of this back end will be:

- Communication to server

- Allowing the user to log in
- Allowing the user to check status of any reports
- Allowing the user to submit a report
- Allowing the user to pull data from the server including messages

These bullet points are what the objective will be measured against. Once this functionality is available in the client, then the objective will have been attained.

This objective is relevant as without the client, users would not be able to submit reports / answer them. This should be attained within a further 5 days of the completion of Objective three.

2.18.5 Objective Five Client Front end

My fourth SMART objective will be to create the client side Front end. This is important as most users of the program will outright refuse to use this if it doesnt have a GUI. This will also make the program much easier to use for younger students who may be less competent with computers.

This objective will be measured against the opinions of both my end users, and other students within the school whom I will send beta clients to. This will be attained once clients cannot give beneficial feedback and when all bugs are removed from the client.

This objective, whilst cosmetic, is vital to ensure the program is used by all students efficiently, as lots of students do not want to report bullying, so they may look for any flaws in the program to give them some cover to hide behind I wanted to report it but the program would not let me I tried but the program was too hard

The objective should be completed within 5 days of the completion of objective Four.

2.18.6 Objective Six - Web Page and Web Client

My Fifth SMART objective will be to create a web page advertising my program, as well as hosting a download link. The web page will contain all relevant information regarding the program including users feedback and an embedded demo video. The web page should also contain a web-client on a sub domain e.g. Web-client.Secureport.space from which users can submit anonymous reports.

This Objective will be met once the web page is live and once the web client contains relevant functionality. Once this is true, the objective will have been attained.

This objective is relevant as users will need a place to download the file from, otherwise they cannot possible use the clients. This should be attained within a further 5 days of the completion of Objective Five.

2.18.7 *Optional* Objective Seven - Further functionality

This Objective is not relevant to the main program as it does not fall into the critical path, however it is in the non-critical path so I have included it as an optional objective.

Here is what is written in the non critical path, which is what Objective Seven focuses on implementing.:

- Email functionality
- Extend web client to full functionality.
- ENCRYPT EVERYTHING

I will split objective seven up into three sub-objectives that are each small objectives that can be individually completed.

The first objective of objective Seven, hereby Objective 7.1 is To add email functionality.

The second objective of objective seven, hereby objective 7.2 is to extend the functionality of the web client to that of the desktop client.

The final objective of objective seven, hereby objective 7.3 is to encrypt everything.

2.18.7.1 Objective 7.1

The purpose of Objective 7.1 is To add email functionality to the report status. This will be during the report submission stage, when a confirmation email is sent to show that the report has been seen by a teacher and that they are working on it. The program will also send an email when a teacher replies to the report, and when it is marked as resolved.

2.18.7.2 Objective 7.2

The purpose of Objective 7.2 is To extend the functionality of the web client to that of the desktop client, that means extending it so that the web client:

- can run with stablility
- can log in
- can send pull / push requests to server
- can check status of previous reports
- view past message history, view details and end results

2.18.7.3 Objective 7.3

The purpose of Objective 7.3 is To Encrypt everything using a *strong encryption algorithm. Everything will be encrypted including transfer between clients and server,passwords in databases -salted and hashed as opposed to encrypted, messages stored in databases

Chapter 3

Design

Within this design section I will describe how I intent to structure my program, and how it will be developed.

3.1 My end users

In this subsection I will make an appreciation of how the end users will influence the design of my program.

3.1.1 how my end users will access the program

Due to my project being a web application, I will host the application on a permanent domain that can be freely accessed in any location. This enables any device to connect and submit a report for the web application. This will also work both within the school and at home, and is a more usable solution that a client that requires downloading as that may have incompatibility with home users if it is designed specifically for the environment (windows 10) at school, as some students may use OSX / linux devices with would not be suported by default.

From consideration of my research into different user interfaces, I have decided to make a dark themed application as this improves readability and accessibility. This is important as all students have different levels of ability and technical knowledge. As such my program should be easy to see and understand for all users regardless of skill.

I have mocked up a design for the main page of my program to demonstrate this. Which can be seen in the appendices, here: 11

This is a very simple interface which fits perfectly with my research in section 2.5.1 "An Appreciation of the skills possessed by my end users"

For the program I have also mocked up a flowchart that models how the report will be submitted. This can be seen in the appendices here: 12

As you can see the flow chart is split into different sections for each stage of the program, with the exception of the end user (student). This makes the model easy to understand as it runs through what each stage of my program will have to do, ensuring that only valid data is processed and stored. From my research into GDPR, maintaining data validity and integrity is a key focus of the new regulation, and taking this step to check its valid will enable my program to be compliant with the new regulation.

3.1.2 code and organisation

For my project I will use the web language JavaScript combined with JQuery.

The file structure of which will be organised into two folders, one for the the student page, and one for the admin page. Each of these folders will have an html file, css file, and js file for the body , cosmetics and logic respectively.

3.2 How the solution will be developed

The project will be created and will be hosted online, with a series of prototypes that will be explained to my end user in order to get feedback. As a result of this the structure will be easily tracked, and each prototype will be saved and stored Version control system such as git.

The impact this has upon my end user of mr pilbury, is that he will be able to track progress and ensure that the program will be suitable for use within the school.

This will also impact the students that ultimately use the program, as the solution i will be able to develop will be made more efficient, usable and functional based on feed back provided during the prototyping phase of development.

Chapter 4

Implementation

Within this section I will discuss all things code. The Full code can be seen within the appendices.

4.1 structure of my code

as my project is a web application it is built up of many scripts and files within a hosted directory.

4.1.1 File directory view

this is the directory of my application:

[public]
-404.html
-index.html
-favicon.ico
-script.js
-style.css
-sweetalert2.all.min.js

—[admin]
— -sweetalert2.all.min.js

— -favicon.ico
— -index.html
— -script.js
— -style.css

each line is a file prepended by - directories are marked by square brackets and all files within the directory are marked by — -

as you can see from the file structure there is a public folder, which is hosted on the Google firebase hosting platform. This can be accessed via DOMAIN.EXT within that there are 6 files and another directory, admin, which contains the files for the admin page which can be accessed via DOMAIN.EXT/admin however requires a sign in.

There are screen shots of the directory of public here: 13 and one for admin here: 14 within the appendices.

My full code can be found in the appendices and is split up into the directory structure shown above. However, will be omitting the sweetlalert2.all.min.js file as this is an open source library for replacing the JavaScript alerts. This was important to include as it enables my program to be very accessible to students who may be uneasy about the default JavaScript alert, however I established that it would be unwise to write my own script for this as it would reduce the time I would have for programming the rest of the application.

here is the full code for my program split into the files described above:

Public

404.html is an automatically generated file, created by firebase hosting, it is the default page that a user is directed to when the server detects a 404 error (page not found). I could have created my own 404 page however I established that it would be unwise to write my own as it would reduce the time I would have for programming the rest of the application.

```
<!DOCTYPE html>
<html>
```

```
<head>
  <meta charset="utf-8">
  <meta name="viewport"
  content="width=device-width, initial-scale=1">
  <title>Page Not Found</title>

  <style media="screen">
    body { background: #ECEFF1; color: rgba(0,0,0,0.87);
    font-family: Roboto,
    Helvetica, Arial, sans-serif; margin: 0; padding: 0; }
    #message { background: white; max-width: 360px; margin:
    100px auto 16px;
    padding: 32px 24px 16px; border-radius: 3px; }
    #message h3 { color: #888; font-weight: normal;
    font-size: 16px; margin:
    16px 0 12px; }
    #message h2 { color: #ffa100; font-weight: bold;
    font-size: 16px; margin:
    0 0 8px; }
    #message h1 { font-size: 22px; font-weight: 300;
    color: rgba(0,0,0,0.6);
    margin: 0 0 16px;}
    #message p { line-height: 140\%; margin: 16px 0 24px;
    font-size: 14px; }
    #message a { display: block; text-align: center;
    background: #039be5;
    text-transform: uppercase; text-decoration: none;
    color: white;
    padding: 16px; border-radius: 4px; }
    #message, #message a { box-shadow: 0 1px
    3px rgba(0,0,0,0.12),
    0 1px 2px rgba(0,0,0,0.24); }
    #load { color: rgba(0,0,0,0.4); text-align:
    center; font-size: 13px; }
    @media (max-width: 600px) {
      body, #message { margin-top: 0; background:
```

```
        white; box-shadow: none; }
        body { border-top: 16px solid #ffa100; }
      }
    </style>
  </head>
  <body>
    <div id="message">
      <h2>404</h2>
      <h1>Page Not Found</h1>
      <p>The specified file was not found on this website.
      Please check the URL for mistakes and try again.</p>
      <h3>Why am I seeing this?</h3>
      <p>This page was generated by the Firebase Command-Line Interface.
      To modify it, edit the <code>404.html</code> file in your project's
      configured <code>public</code> directory.</p>
    </div>
  </body>
</html>
```

index.html is the main HTML file for the public directory. The first 30 lines are importing initial library's hosted on Google. The first is a font, which is used to improve the accessibility and readability of my site, this is important as the program may be used by students who are visually impaired and as such, steps should be taken to ensure they can benefit from the functionality of my program.

The second link is to my stylesheet, written in CSS, which manipulates the html and applies visual effects to it, such as changing colour of the background and text. This is important as it enables me to move things around the screen to make the program much more easily navigated.

the next script is imported from google. The first being jQuery which is a JavaScript framework which extends the functionality of default JavaScript whilst also making it faster to develop in. JQuery also creates some functionality for CSS manipulation and animation, which I have used to close one of the divs in the program, so it slides down and the webpage resizes around it, meaning when it is closed, there is more space for the rest of the functionality.

the next two scripts are also imported from google and are the firebase libraries, which enable email authentication as well as the database features. This is important as without either of these my program would not be able to function.

The last script in the html head is where I initialise the local firebase session, and link it to the database so that it can both authenticate users as well as store and retrieve reports from the database.

within the body of the html file is where most of the web content is defined. Firstly, there is a login div, which is first shown when the user enters the page for the first time. This is then hidden when the user either logs in with an email and password, or if they choose to submit a report anonymously. It is hidden through CSS manipulation in my script.js file, which I will explain later.

mostly the html file is self-explanatory and does not need any more explanation as it is comprised of divs that work the same as the login div explained above. However, there are two divs that I would like to spend some time discussing. These are the student-report div and the get-help div.

firstly I will explain the student-report div. This div is comprised of a form. The form within the html file alone does not do much, it takes user inputs into the respective fields, which are easily viewed, and then allows a user to submit the report once all fields are filled. If a field is not filled, the user is warned of this due to the use of the required parameter. The form is monitored by the Javascript within script.js and when the submit button is clicked, the javascript code parses this to the database. I will go into more detail about this when I cover the script.js code.

Finally the get-help div is a message box that pops up, offering some suggestions about bullying, and provides some links to bullying information as well as some free charity run phone lines; if the student is being bullied and requires immidiate help.

```
    <html>
<head>
    <meta charset="UTF-8">
    <title>Secureport Student</title>

    <link href="https://fonts.googleapis.com/css?family=
```

```
    Nunito:400,600,700" rel="stylesheet">
    <!--
    <link rel="stylesheet" href="https://maxcdn.
    bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css">-->
    <link rel="stylesheet" href="./style.css">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery
    /3.3.1/jquery.min.js"></script>
    <script src="https://www.gstatic.com/firebasejs/4.12.1/
    firebase.js"></script>
    <script src="https://www.gstatic.com/firebasejs/4.12.1/
    firebase-firestore.js"></script>

    <script>
        // Initialize Firebase
        var config = {
            apiKey: "AIzaSyD1a920sL_Dzd1herQzUfnaeR5NIIsOSzM",
            authDomain: "port-a6eda.firebaseapp.com",
            databaseURL: "https://port-a6eda.firebaseio.com",
            projectId: "port-a6eda",
            storageBucket: "port-a6eda.appspot.com",
            messagingSenderId: "515080856207"
        };
        firebase.initializeApp(config);
    </script>



</head>
<body>

  <div id="login_div" class="main-div">
    <h3>SecuReport Web Login</h3>
    <input type="email" placeholder="Email..." id="email_field"/>
    <input type="password" placeholder="Password..." id="password_field" />
```

```
    <button onclick="login()">Login to Account</button>




  </div>
    <div id="anon_login-div" class="anon_main-div">
        <button onclick="login_anon()" >Submit report Anonymously</button>

   </div>

  <div id = "anon_user_div" class="loggedin-div">
      <h3>Welcome User</h3>
      <p>Welcome to a test version of secureport. You're currently
      logged in anonymously.</p>
      <p> This means that unfortunately you cannot view previous
      reports, but you can submit them regardless</p>
        <button onclick="logout()">Return to login</button>
  </div>

  <div id="user_div" class="loggedin-div">
    <h3>Welcome User</h3>
    <p id="user_para">Welcome to a test version of secureport.
    You're currently logged in.</p>
    <button onclick="logout()">Logout</button>
  </div>
<div class="container">
<div id="student-report" class="report-form student-form">

    <form id="newActivity">
    <header>
        <h3>Submit a new Report</h3>
        <div>Tell us what happened!</div>
        <p></p>
    </header>
    <div>
```

```
        <label class="desc" id="title1" for="Field1">Title: </label>
        <div>
            <input id="activityTitle" name="Field1" type="text"
            value="" tabindex="1" placeholder="brief summary of
            issue" required>
        </div>
    </div>
    <div>
        <label class="desc" id="description1" for="Field2">
        Description: </label>
        <div>
            <textarea id="activityDescription" name="Field2"
            spellcheck="true" rows="10" col="50" tabindex="2"
            placeholder="main body text" required></textarea>
        </div>
    </div>

    <div>
        <button id="saveForm" name="saveForm" type="submit">
        Submit Report</button>
    </div>

</form>
</div>

</div>

<div id="previous-reports" class="student-view">


</div>

</body>
<footer>
```

```
      <div id = "get-help" class="fragment">
          <span id='close'>X</span>
        <h3>Need help sooner?</h3>
        <p>There are many websites which can offer help immediately
        so you can feel happier in yourself and put the bullying
        behind you.</p>
        <a href="https://www.ditchthelabel.org/cyber-
        bullying-top-9-tips-on-overcoming-it/"> Cyber bullying tips </a>
        <br>
        <a href="https://www.childline.org.uk/info-advice/
        bullying-abuse-safety/types-bullying/">
        Types of bullying</a>
        <br>
        <a href="https://www.ditchthelabel.org/bullying-101/
        "> Bullying 101</a>
        <p>The National Bullying Helpline |
        CALL : 0845 22 55 787 | EMAIL : admin@national
        bullyinghelpline.co.uk</p>

      </div>

</footer>
<script src="script.js"></script>
<script src="sweetalert2.all.min.js"></script>
</html>



var anonuser = false;
var loggedin = false;
var show = "block";
var hide = "none";
var db = firebase.firestore();
var uid;

firebase.auth().onAuthStateChanged(function(user) {
  if (user) {
```

```
    // User is signed in.
    loggedin = true;

    document.getElementById("user_div").style.display = show;
    document.getElementById("login_div").style.display = hide;
    document.getElementById("anon_login-div").style.display = hide;
    document.getElementById("student-report").style.display= show;
    document.getElementById("previous-reports").style.display= show;

    \$(document.getElementsByClassName("fragment")).hide();

    var user = firebase.auth().currentUser;
      var email = user.email;
    uid = user.uid;
    console.log(email);
    console.log(uid);


      if  (email == null) {

          anonuser = true;
          document.getElementById("anon_user_div").style.display= show;
        document.getElementById("previous-reports").style.display= hide;
          document.getElementById("user_div").style.display = hide;

          // user is anonymous
          // save authData.uid to some local session info,
          to keep track of data

      } else {
          anonuser = false;
          document.getElementById("anon_user_div").style.display= hide;
      // user is logged in
      // transfer any data that you had stored from the
      // anonymous session to this new authUser.uid
  }
```

```
    if(user != null){

      var email_id = user.email;
      document.getElementById("user_para").innerHTML =
      "Welcome : "+ email_id;

    }

  } else {
    // No user is signed in.
    loggedin = false;
    document.getElementById("user_div").style.display = hide;
    document.getElementById("login_div").style.display = show;
    document.getElementById("anon_login-div").style.display = show;
    document.getElementById("student-report").style.display= hide;
    document.getElementById("previous-reports").style.display= hide;
    document.getElementById("anon_user_div").style.display= hide;


}
});
function login_anon(){
    firebase.auth().signInAnonymously().catch(function(error) {
  // Handle Errors here.
  var errorCode = error.code;
  console.log(errorCode);
  var errorMessage = error.message;
  console.log(errorMessage)

  // ...
});
};

function login(){

  var userEmail = document.getElementById("email_field").value;
```

```
  var userPass = document.getElementById("password_field").value;

  firebase.auth().signInWithEmailAndPassword(userEmail, userPass).
            catch(function(error) {
    // Handle Errors here.
    var errorCode = error.code;
    var errorMessage = error.message;
    swal({
        title: 'Error!',
        text: errorMessage,
        type: 'error',
        confirmButtonText: 'Dont worry I\'ll sort it now xoxo'

    })
    // ...
  });


}

function logout(){
  firebase.auth().signOut();
  loggedin = false;
}


\$('#newActivity').submit(function(event) {
    var \$form = \$(this);
    console.log("submit to Firebase");


    //make the submit disabled
    \$form.find("#saveForm").prop('disabled', false);

    //get the actual values that we will send to firebase
    var titleToSend = \$('#activityTitle').val();
```

```
    console.log(titleToSend);

    var descriptionToSend = \$('#activityDescription').val();

    console.log(descriptionToSend);


    //take the values from the form, and put them in an object
    var date = new Date();
    var unixtime = date.getTime();
    console.log(unixtime);

    console.log(uid);
    db.collection('test').doc(String(unixtime)).set({
        "description": descriptionToSend,
        "title": titleToSend,
        "status": "001",
        "resolution": "",
        "user": uid
    })
        .then(function (docRef) {
            console.log("Document written with ID: ", docRef.id);
        })
        .catch(function (error) {
            console.error("Error adding document: ", error);
        });

    return false;
});

\$(document).ready(function(){
    \$(document.getElementById("close")).click(function(){
        \$(document.getElementsByClassName("fragment")).fadeOut();
        document.getElementById("login_div").style.marginTop = "200px";
    });
```

```
});


\$("#previous-reports").height(\$("#student-report").height());

function addTable() {

    var myTableDiv = document.getElementById("previous-reports");
    var table = document.createElement('TABLE');
    var tableBody = document.createElement('TBODY');

    table.border = '1';
    table.appendChild(tableBody);

    var heading = new Array();
    heading[0] = "Title";
    heading[1] = "description";
    heading[2] = "resolution";

    var report = new Array()

    db.collection("test").get().then(function (querySnapshot) {
        var x = 0;
        querySnapshot.forEach(function (doc) {
            doc.data().user, uid);
            if (doc.data().user == uid) {
                console.log(doc.id, " => ", doc.data());
                report[x] = new Array(doc.data().title,
                doc.data().description, doc.data().resolution);
                x += 1;
            }

            else {
                x = x;
            }
```

```
        });
    })
        .catch(function (error) {
            console.log("Error getting documents: ", error);
        });

    //TABLE COLUMNS
    var tr = document.createElement('TR');
    tableBody.appendChild(tr);
    for (i = 0; I < 3; i++) {
        var th = document.createElement('TH');
        th.width = '125';
        th.appendChild(document.createTextNode(heading[i]));
        tr.appendChild(th);
    }

    //TABLE ROWS
    console.log("report length: ",report.length);
    for (i = 0; I < report.length; i++) {
        var tr = document.createElement('TR');
        tableBody.appendChild(tr);
        for (j = 0; j < 3; j++) {
            var td = document.createElement('TD');
            td.width = '125';
            console.log(report[i][j]);
            td.appendChild(document.createTextNode(report[i][j]));
            tr.appendChild(td)
        }
        tableBody.appendChild(tr);
    }
    myTableDiv.appendChild(table);
}

addTable();
```

```
    *{
overflow: hidden;
}

body {
  background: #515151;
  padding: 0px;
  margin: 0px;
  font-family: 'Nunito', sans-serif;
  font-size: 16px;
color: #fff;
}

input, button {
  font-family: 'Nunito', sans-serif;
  font-weight: 700;
}

.main-div, .loggedin-div {
  width: 65\%;
  margin: 0px auto;
  margin-top: 105px;
  padding: 0px;
  display: block;
  max-width: 350px;
  transition: all 2s;
  -webkit-transition: all 2s;
}
.anon_main-div{
  width: 65\%;
  margin: 0px auto;
  padding: 0px;
  display: block;
  max-width: 350px;
}
```

```
.main-div input,.main-div button,.anon_main-div button,
    .loggedin-div button{
  display: block;
  border: 1px solid #ccc;
  border-radius: px;
  background: #616161;
  padding: 15px;
  outline: none;
  width: 100\%;
  color: #FFFFFF;
  margin-bottom: 10px;
  transition: 0.3s;
  -webkit-transition: 0.3s;
  -moz-transition: 0.3s;
}
.loggedin-div button{
  width: 40\%;
}


.main-div button{
  border: 1.5px solid #5d8ffc;
}
.anon_main-div button{
border: 1.5px solid #FFB200;
}
.main-div input:focus {
  border: 1px solid #777;
}
#brief-advice{
  padding: 0px;
  margin: 0px;
  font-family: 'Nunito', sans-serif;
  font-size: 12px;
  color: #fff;
  position: absolute;
```

```
  width: 50\%;
  height: 150px;
  z-index: 15;
  top: 50\%;
  left: 5\%;
  margin-top: 200px;
}


.main-div button:hover, .loggedin-div button:hover {
  background: #515151;
  color: #5d8ffc;
  border: 1.5px solid #FFB200;
  cursor: pointer;
}


.anon_main-div button:hover{
  background: #515151;
  color: #FFB200;
  border: 1.5px solid #5d8ffc;
  cursor: pointer;
}

.swal2-popupl{
background: #515151 !important;
color: #FFFFFF;
}

.fragment {
    background: #616161;
    height: 23\%;
    border: 1px solid #ccc;
    border-radius: 3px;
    display: block;
    padding: 10px;
```

```
    padding-top: 5px;
    padding-right: 10px;
    box-sizing: border-box;
    text-decoration: none;
    margin: 10px;
    margin-top: 15\%;

}

.fragment:hover {
    box-shadow: 2px 2px 5px rgba(0,0,0,.2);

}

#close {
    float:right;
    display:inline-block;
    padding:0.2\% 0.5\%;
    color: #000;
    background:#ccc;
}

#close:hover {
        background:#ccc;
    color:#fff;
    }


*{
  padding: 2px;
}body {
  background: #515151;
  padding: 0px;
  margin: 0px;
  font-family: 'Nunito', sans-serif;
  font-size: 16px;
```

```
color: #fff;
}

input, button {
  font-family: 'Nunito', sans-serif;
  font-weight: 700;
}
textarea{
  width: 40\%;
  resize: none;

}
.loggedin-div {
    top: -10\%;
    display: none;
    width: 45\%;
    margin-left: 7\%;
}

.report-form {
  position: fixed;
  display: none;
  width: 50\%;
  margin-left: -15\%;
  margin-top: 50px;
  margin-right: 200px;
  padding: 10px;
  max-width: 600px;
}
.report-form input,#activityDescription, .report-form button{
  top:150px;
  display: block;
  border: 1px solid #ccc;
  border-radius: 2px;
  background: #616161;
  padding: 15px;
```

```
  outline: none;
  width: 100\%;
  color: #FFFFFF;
  margin-bottom: 20px;
  transition: 0.3s;
  -webkit-transition: 0.3s;
  -moz-transition: 0.3s;
}

.report-form input:focus {
  border: 1px solid #777;
}

./report-form button, ./loggedin-div button{

  margin:  auto;
  background: #5d8ffc;
  color: #515151;
  border: 1.5px solid #FFB200;
  border-radius: 5px;
  padding: 15px;
  width: 60\%;
  transition: 0.3s;
  -webkit-transition: 0.3s;
  -moz-transition: 0.3s;
}

.report-form button:hover, .loggedin-div button:hover {
  background: #515151;
  color: #5d8ffc;
  border: 1.5px solid #FFB200;
  cursor: pointer;
}

.anon_report-form button{
  background: #FFB200;
```

```
  color: #515151;
  border: 1.5px solid #5d8ffc;
  border-radius: 5px;
  padding: 15px;
  display: block;
  width: 65\%;
  transition: 0.3s;
  -webkit-transition: 0.3s;
  -moz-transition: 0.3s;
}

.anon_report-form button:hover{
  background: #515151;
  color: #FFB200;
  border: 1.5px solid #5d8ffc;
  cursor: pointer;
}

.swal2-popupl{
background: #515151 !important;
color: #FFFFFF;
}
#anon_user_div{
  width:50\%;
  position:fixed;
  top:100px;
  left:60\%;
  max-height: 480px;
}
#anon_user_div button{
  width: 100\%

}
#previous-reports {
    position:fixed;
    top:360px;
```

```
    left:60\%;
    max-height: 480px;
}
#previous-reports{
    align-items: center;
    padding: 5px;
        background-color: #515151;
        align-content: center;
    }
#previous-reports tr {
    height: 45px;
    padding: 20px;
    border-bottom: 1px solid #ccc;
}
#previous-reports td{
    width: 200px;

}
```

Admin

```
    <html>
<head>
    <meta charset="UTF-8">
    <title>Secureport Admin</title>
    <link href="https://fonts.googleapis.com/css?
    family=Nunito:400,600,700" rel="stylesheet">
    <script src="https://ajax.googleapis.com/ajax/
    libs/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://www.gstatic.com/firebasejs
    /4.12.1/firebase.js"></script>
    <script src="https://www.gstatic.com/firebasejs
    /4.12.1/firebase-firestore.js"></script>
    <script>
        // Initialize Firebase
```

```
        var config = {
            apiKey: "AIzaSyD1a920sL_Dzd1herQzUfnaeR5NIIsOSzM",
            authDomain: "port-a6eda.firebaseapp.com",
            databaseURL: "https://port-a6eda.firebaseio.com",
            projectId: "port-a6eda",
            storageBucket: "port-a6eda.appspot.com",
            messagingSenderId: "515080856207"
        };
        firebase.initializeApp(config);
    </script>
    <link rel="stylesheet" href="./style.css" />
</head>
<body>

  <div id="login_div" class="main-div">
    <h3>SecuReport Web Login</h3>
    <input type="email" placeholder="Email..."
    id="email_field"/>
    <input type="password" placeholder="Password..."
    id="password_field" />

    <button onclick="login()">Login to Account</button>




  </div>
    <div id="anon_login-div" class="anon_main-div">
        <button onclick="login_anon()" >Submit report
        Anonymously</button>

   </div>
  <div id="user_div" class="loggedin-div">
    <h3>Welcome User</h3>
    <p id="user_para">Welcome to a test version of
    secureport.You're currently logged in.</p>
```

```
    <button onclick="logout()">Logout</button>
  </div>


  <div id = admin-data>
      <p id="data"></p>
  </div>



</body>
<footer>


</footer>

    <script src="script.js"></script>
    <script src="sweetalert2.all.min.js"></script>
</html>




var db = firebase.firestore();

firebase.auth().onAuthStateChanged(function(user) {
  if (user) {
    // User is signed in.

    document.getElementById("user_div").style.display = "block";
    document.getElementById("login_div").style.display = "none";
    document.getElementById("anon_login-div").style.display = "none";
    document.getElementById("admin-data").style.display= "block";
    \$(document.getElementsByClassName("fragment")).hide();

    var user = firebase.auth().currentUser;
```

```
    if(user != null){

      var email_id = user.email;
      document.getElementById("user_para").innerHTML =
      "Welcome : " + email_id;

    }

  } else {
    // No user is signed in.

    document.getElementById("user_div").style.display = "none";
    document.getElementById("login_div").style.display = "block";
    document.getElementById("anon_login-div").style.display = "block";
    document.getElementById("admin-data").style.display= "none";


}
});
function login_anon(){
    firebase.auth().signInAnonymously().catch(function(error) {
  // Handle Errors here.
  var errorCode = error.code;
  var errorMessage = error.message;
  // ...
});
}
function login(){

  var userEmail = document.getElementById("email_field").value;
  var userPass = document.getElementById("password_field").value;

  firebase.auth().signInWithEmailAndPassword
  (userEmail, userPass).catch(function(error) {
    // Handle Errors here.
```

```
    var errorCode = error.code;
    var errorMessage = error.message;
    swal({
        title: 'Error!',
        text: errorMessage,
        type: 'error',
        confirmButtonText: 'Dont worry I\'ll sort it now xoxo'
    })
    // ...
  });


}

function logout(){
  firebase.auth().signOut();
}


db.collection("test").where("status", "==", "001")
    .get()
    .then(function (querySnapshot) {
        querySnapshot.forEach(function (doc) {
           // doc.data() is never undefined for query doc snapshots
            console.log(doc.id, " => ", doc.data());
            document.getElementById("data").innerHTML = string(doc.id
            + " => ") + doc.data().title + " || " + doc.data().description;
        });
    })
    .catch(function (error) {
        console.log("Error getting documents: ", error);
    });


\$(document).ready(function(){
    \$(document.getElementById("close")).click(function(){
```

```
        \$(document.getElementsByClassName("fragment")).fadeOut();
        document.getElementById("login_div").style.marginTop = "200px";
    });
});
```

```
    *{
overflow: hidden;
}

body {
  background: #515151;
  padding: 0px;
  margin: 0px;
  font-family: 'Nunito', sans-serif;
  font-size: 16px;
color: #fff;
}

input, button {
  font-family: 'Nunito', sans-serif;
  font-weight: 700;
}

.main-div, .loggedin-div {
  width: 65\%;
  margin: 0px auto;
  margin-top: 105px;
  padding: 0px;
  display: block;
  max-width: 350px;
  transition: all 2s;
  -webkit-transition: all 2s;
}
.anon_main-div{
```

```
  width: 65\%;
  margin: 0px auto;
  padding: 0px;
  display: block;
  max-width: 350px;
}

.main-div input,.main-div button,.anon_main-div button, .loggedin-div button{
  display: block;
  border: 1px solid #ccc;
  border-radius: px;
  background: #616161;
  padding: 15px;
  outline: none;
  width: 100\%;
  color: #FFFFFF;
  margin-bottom: 20px;
  transition: 0.3s;
  -webkit-transition: 0.3s;
  -moz-transition: 0.3s;
}
.main-div button{
  border: 1.5px solid #5d8ffc;
}
.anon_main-div button{
border: 1.5px solid #FFB200;
}
.main-div input:focus {
  border: 1px solid #777;
}
#brief-advice{
  padding: 0px;
  margin: 0px;
  font-family: 'Nunito', sans-serif;
  font-size: 12px;
  color: #fff;
```

```
  position: absolute;
  width: 50\%;
  height: 150px;
  z-index: 15;
  top: 50\%;
  left: 5\%;
  margin-top: 200px;
}


.main-div button:hover, .loggedin-div button:hover {
  background: #515151;
  color: #5d8ffc;
  border: 1.5px solid #FFB200;
  cursor: pointer;
}


.anon_main-div button:hover{
  background: #515151;
  color: #FFB200;
  border: 1.5px solid #5d8ffc;
  cursor: pointer;
}

.swal2-popupl{
background: #515151 !important;
color: #FFFFFF;
}

.fragment {
    background: #616161;
    height: 23\%;
    border: 1px solid #ccc;
    border-radius: 3px;
    display: block;
```

```
    padding: 10px;
    padding-top: 5px;
    padding-right: 10px;
    box-sizing: border-box;
    text-decoration: none;
    margin: 10px;
    margin-top: 15\%;

}

.fragment:hover {
    box-shadow: 2px 2px 5px rgba(0,0,0,.2);

}

#close {
    float:right;
    display:inline-block;
    padding:0.2\% 0.5\%;
    color: #000;
    background:#ccc;
}

#close:hover {
        background:#ccc;
    color:#fff;
    }



*{
  padding: 2px;
}body {
  background: #515151;
  padding: 0px;
  margin: 0px;
  font-family: 'Nunito', sans-serif;
```

```
  font-size: 16px;
color: #fff;
}

input, button {
  font-family: 'Nunito', sans-serif;
  font-weight: 700;
}
textarea{
  width: 40\%;
  resize: none;

}
.report-form, .loggedin-div {
  display: none;
  width: 85\%;
  margin-left: 5\%;
  margin-top: 200px;
  margin-right: 200px;
  padding: 10px;
  max-width: 1600px;
}
.report-form input,#activityDescription, .report-form button{
  display: block;
  border: 1px solid #ccc;
  border-radius: 2px;
  background: #616161;
  padding: 15px;
  outline: none;
  width: 40\%;
  color: #FFFFFF;
  margin-bottom: 20px;
  transition: 0.3s;
  -webkit-transition: 0.3s;
  -moz-transition: 0.3s;
}
```

```
.report-form input:focus {
  border: 1px solid #777;
}

./report-form button, ./loggedin-div button{

  margin:  auto;
  background: #5d8ffc;
  color: #515151;
  border: 1.5px solid #FFB200;
  border-radius: 5px;
  padding: 15px;
  width: 40\%;
  transition: 0.3s;
  -webkit-transition: 0.3s;
  -moz-transition: 0.3s;
}

.report-form button:hover, .loggedin-div button:hover {
  background: #515151;
  color: #5d8ffc;
  border: 1.5px solid #FFB200;
  cursor: pointer;
}

.anon_report-form button{
  background: #FFB200;
  color: #515151;
  border: 1.5px solid #5d8ffc;
  border-radius: 5px;
  padding: 15px;
  display: block;
  width: 100\%;
  transition: 0.3s;
  -webkit-transition: 0.3s;
```

```
  -moz-transition: 0.3s;
}

.anon_report-form button:hover{
  background: #515151;
  color: #FFB200;
  border: 1.5px solid #5d8ffc;
  cursor: pointer;
}

.swal2-popupl{
background: #515151 !important;
color: #FFFFFF;
}
```

Chapter 5

Testing

This chapter will be on testing my code.

The solution I developed was tested in two different ways: Feedback from students (end users), and a software test plan methodology.

5.1 end users

Here is some of the feedback that was attained through end users testing both the prototypes and final solution, they are anonymous quotes as i have found that this enables me to gather better and more constructive feedback due to the nature of the program, for example the students who are most likely to use the program are more vulnerable and shy, and may fear judgement from using the program if others were to find out they were bullied or trying to report this. :

5.1.1 quotes

"It is well thought out and easy to use website which i am sure to use in the future as it is less awkward than talking to teachers who arent always supportive"

"I think that the program is well made and very easy to use!"

"i have tested the program three times now, and here is what i think of it: the first time i tested it, it was very basic simple and ugly, but the main things that

were needed were there as i could submit a report. Now the program is a lot more visually appealing and styled, And i love that you have included the extra section on getting help immediately from charities!"

5.2 Testing table

The following is the software test plan methodology briefly explained above. This is important to include as it shows a thorough basis for functionality of the solution that i have developed. This also is an easy way to establish and track any bugs within the code, as tests can be repeated after code is modified.

The testing completed was Functional testing, which checks the application to ensure that it is functional and does what it is supposed to do. This was done along side Cross browser testing, i.e. all tests were carried out on three main web-browsers : Firefox, Chrome, and Edge. These three were chosen as firefox and chrome are the two most popular browsers, and edge is known to sometimes act differently to expected. These three are also the most likely for my end user to use, as chrome and edge are installed within school.

test	expected	actual	pass
page accessible online	when user goes to the domain, the page loads	page loads as expected.	Yes
user can log in	when a user attempts to sign in with correct credentials, they are allowed to do so	user signs in as expected.	yes
user can log in anonymously	when a user attempts to sign in anonymously they are allowed to do so	anonymous sign in works as expected	yes
user logs in with wrong email	error given to them stating what has gone wrong	error returned as expected: Error! There is no user record corresponding to this identifier. The user may have been deleted.	yes
user logs in with wrong password	error given to user stating what has gone wrong	error returned as expected: Error! The password is invalid or the user does not have a password.	yes
report can be submitted and retrieved	report submitted to database and can be retrieved	report submitted and retrieved as expected	yes

5.2.1 evidence for tests

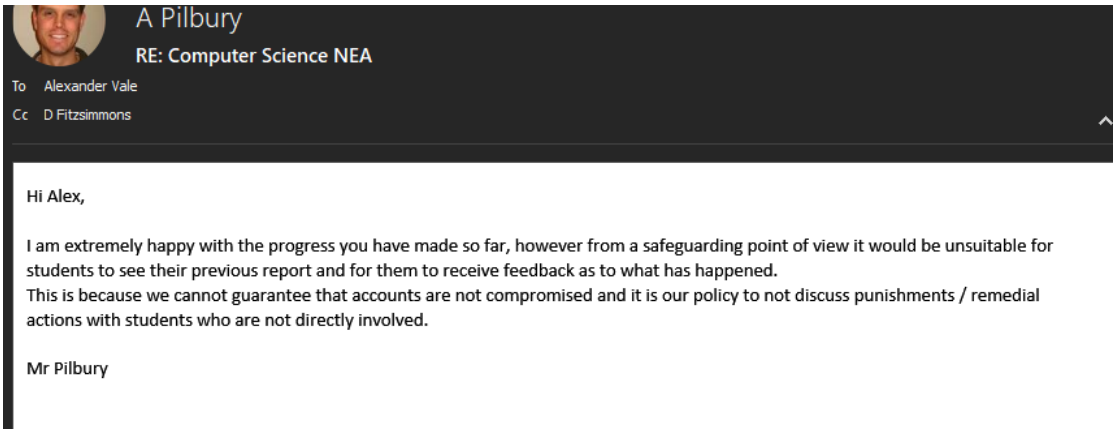
Table 5.1: My caption

test	evidence found in appendices:
page accessible online	appendix 15
user can log in	appendix 16
user can log in anonymously	appendix 17
user logs in with wrong email	appendix 18
user logs in with wrong password	appendix 19
report can be submitted and retrieved	appendix 20

Chapter 6

Evaluation

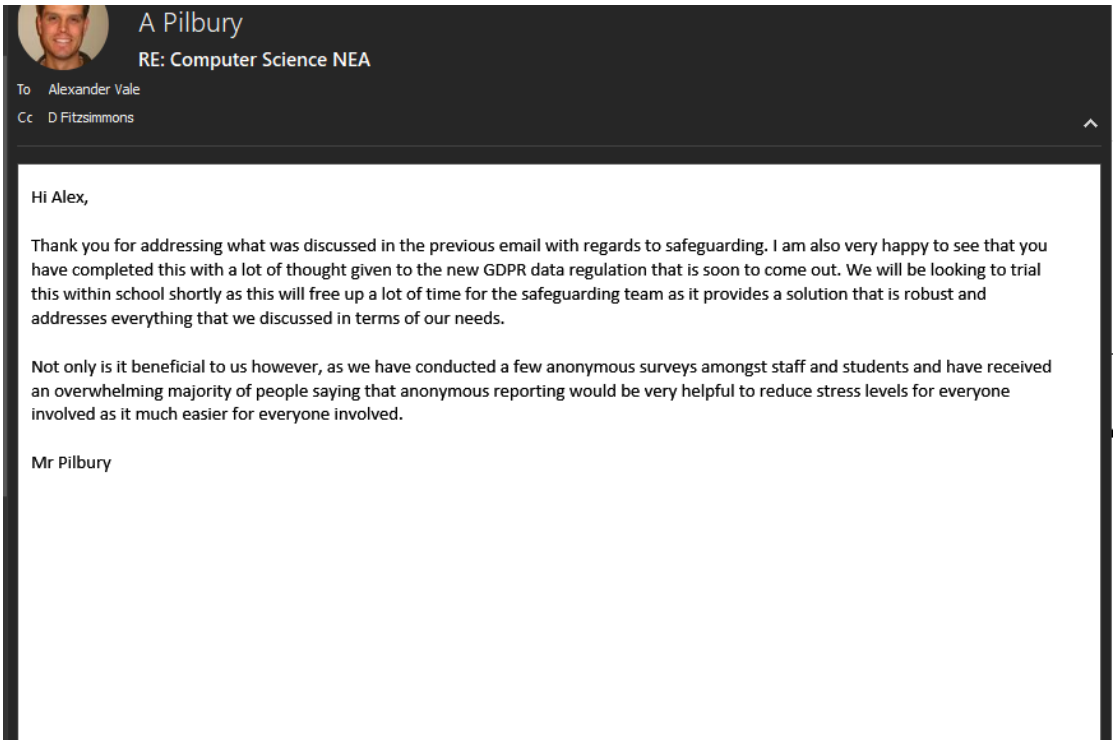
My end user was extremely pleased with the application that I created. This can be seen in the following emails:



This initial feedback email was received after I initially sent off the program to my end user, as this would enable me to get feedback at an early stage, letting me quickly identify which aspects of the program are most crucial to the end user and which are not worth further pursuing.

One example of this is the safeguarding issue that I overlooked, of students viewing report details and the outcome of each report. In hindsight this is fairly obvious that students should not be able to see any further action that has been taken.

As this was spotted early, I could focus my time on other areas of the application and spend time on other areas, such as a complex anonymous reporting offering, which was extremely helpful in the long run, as can be seen in the following email:


A Pilbury
RE: Computer Science NEA
To Alexander Vale
Cc D Fitzsimmons

Hi Alex,

Thank you for addressing what was discussed in the previous email with regards to safeguarding. I am also very happy to see that you have completed this with a lot of thought given to the new GDPR data regulation that is soon to come out. We will be looking to trial this within school shortly as this will free up a lot of time for the safeguarding team as it provides a solution that is robust and addresses everything that we discussed in terms of our needs.

Not only is it beneficial to us however, as we have conducted a few anonymous surveys amongst staff and students and have received an overwhelming majority of people saying that anonymous reporting would be very helpful to reduce stress levels for everyone involved as it much easier for everyone involved.

Mr Pilbury

As you can see in this email, the idea of an anonymous reporting feature is extremely popular among those who were questioned, and possibly would not be included in the application if I did not prototype my application at an early stage.

As a result of this I am very happy as to how the whole process of my program has gone, and if I were to complete another project of a similar scope, I would again include the decision to prototype my application.

6.1 requirments met

The requirements of the project were discussed in section: 2.18 SMART TARGETS.

simplified, these were to:
1) set up a web server 2) set up databases 3) set up server side scripts 4) set up client side scripts 5) complete the interface for the students to see

objectives 1 2 3 were completed through the use of googles Firebase service as this allowed me to ensure complete accesability, as everything hosted is scaled on Googles' Cloud platform and as a result is: faster, more stable, more redundant than anything i could have developed and hosted myself. As a result of using This

both objectives were completed very quickly, and allowed for me to spend more time developing on further requirements.

The fourth and fifth objectives were also completed in parallel due to the nature of web programming, however the interface was initially very basic and only improved later on.

Chapter 7

Conclusion

This is the final chapter of my Report into the program SecuReport.

This will be comprised of two sections: - A section on what I think went well - A section on what I would improve if I were to do it again

7.1 what I think went well

Overall I am very happy with the majority of the report and there are not many things that I would change, however I would like to highlight a few decisions I made which made my life easier and made both the project and report more robust.

7.1.1 Prototyping my application

This is possibly the most important decision I made, and one that I thing is crucial for anyone undertaking a similar project. Without doing any prototyping I would have spend a long time working on features that would only have to be discarded as they were a breach of the safeguarding policy for the school I was developing for. Prototyping allowed me to spot this easily and reallocate the time I would have spent on this section and could further develop features such as anonymous reporting.

7.2 what I would improve if I were to do it again

7.2.1 more prototypes

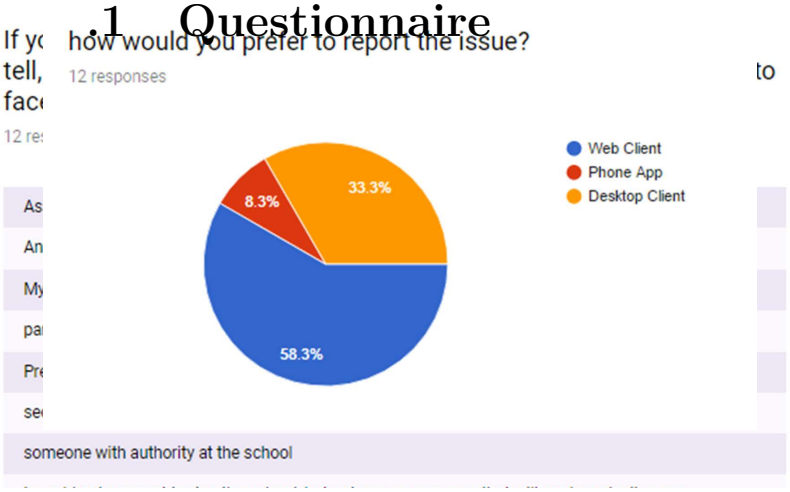
I have already discussed prototyping my application and have highlighted how it enabled me to adjust the time spent on different features, However if I were to Complete a similar project, I would strive to prototype my application sooner and more frequently.

7.2.2 The Cloud

I eventually decided to use Google's cloud platform, firebase, which enabled my program to be more reliable, faster, and have redundancy. I feel this was a crucial step that i overlooked from an early stage, and if i had decided within the analysis to compare cloud platforms rather than examples of web stacks [mean vs lamp] for example and compared firebase and for example AWS then i feel the program would have been developed faster as I spend some development time experimenting with MEAN and LAMP wheras when i decided to use the cloud all my time could be prioritised on development and any low level server maintenance and configuration can be left to google or amazon.

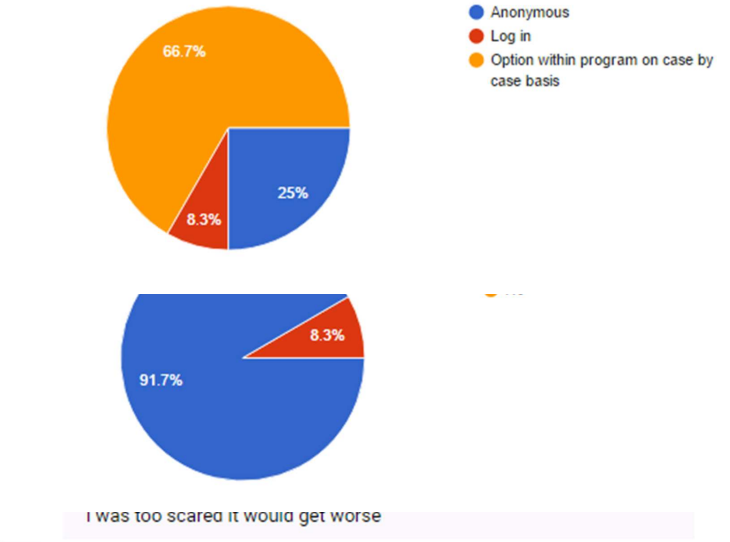
Appendices

.1 Questionnaire



would you like to log in, so you can provide more details of the issue, so it can be resolved more easily, or submit it anonymously? or would you like to be given the choice on each submission?

12 responses



.2 Interviews

.2.1 Teachers

Disclaimer Within this questionnaire, I invite you to participate in a research project regarding the issue of bullying. This study is being done by Alex Vale from the Department of computing at Tytherington School. The purpose of this research study is gather information into the effects of bullying and also the methods of reporting the bullying taken, and preferred method in the future. The survey will take you approximately 10 minutes to complete. Your participation in this study is entirely voluntary and you can withdraw at any time.
Please Briefly outline the current process for reporting bullying, Including an order of events and timeline.
* Long answer text
Please Suggest some improvements that you would make to the process easier for both staff and students.
* Long answer text
Please explain briefly what happens after the report comes in, and the steps the school takes to remediate the actions.
* Long answer text
Within the reporting process, is a time period ever established? For example, if a parent emails the school, is there a maximum time that you aim to have replied by?
* Long answer text
What data do you collect when a report is submitted? How do you store it?
* Long answer text
If a new system was made, what would your requirements be, making sure the program fills all your needs, so it would be used?
* Long answer text

.2.2 Students

Alex Vale - Computer Science NEA Research

Disclaimer Within this questionnaire, I invite you to participate in a research project regarding the issue of bullying. This study is being done by Alex Vale from the Department of computing at Tytherington School. The purpose of this research study is gather information into the effects of bullying and also the methods of reporting the bullying taken, and preferred method in the future. The survey will take you approximately 10 minutes to complete. Your participation in this study is entirely voluntary and you can withdraw at any time.

Please Briefly outline the current process for reporting bullying, from your perspective, from the initial thought of reporting the act, through to the resolution of the situation.

* Long answer text

Please Suggest some improvements that you would make to the process both more easily approachable and comfortable.

* Long answer text

Please explain briefly what feedback you receive / expect to receive from the teachers alongside any investigation that they lead.

* Long answer text

Would you feel it beneficial if someone gave you a time period that you could expect them to stick to, from initial reporting, to the resolution of the issue? if so what would you expect the time period to be?

1. Would NOT find it beneficial
2. would find it beneficial and would expect under two days
3. would find it beneficial and would expect two to five days
4. would find it beneficial and would expect under one week (7 days)
5. would find it beneficial and would expect under a fortnight (14 days)

If a new system was made to make the whole reporting system easier, what would your requirements be, making sure the program fills all your needs and making it appeal to you so you may use it in the future?

Long answer text

.3 Email To Mr Pilbury

**

Dear Mr Pilbury,

For my studies in AQA A-Level computer science, I am required to do an NEA. For this project we are required to create a full computer system which can have

an Impact on the world. We then have research this project, in terms of an end user who would theoretically use the program.

For my program I have chosen to create a system that allows students, as well as parents / carers to securely report bullying, or any worries they may have, to someone that they are both comfortable with, but also who can deal with the problem efficiently. As you are Assistant head, with heavy focus on pastoral care, you would be an excellent end user in this regard. Can you confirm that you are able and willing to be the end user for this project?

Many thanks,
Alex Vale

**

**

Hi Alex,

I am happy to help you with this and Mr Fitzsimmons may also be very good to work with as well

Mr Pilbury

**

.4 UI Design

.4.1 Applications used inside school environment

.4.2 Applications used outisde of school environment

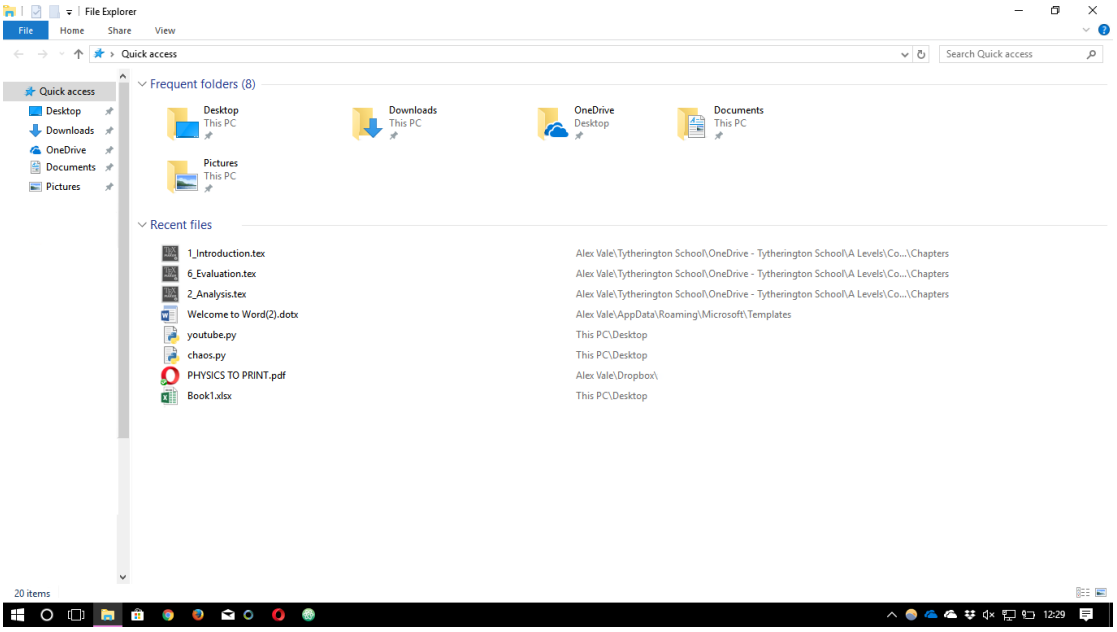


FIGURE 1: File Explorer

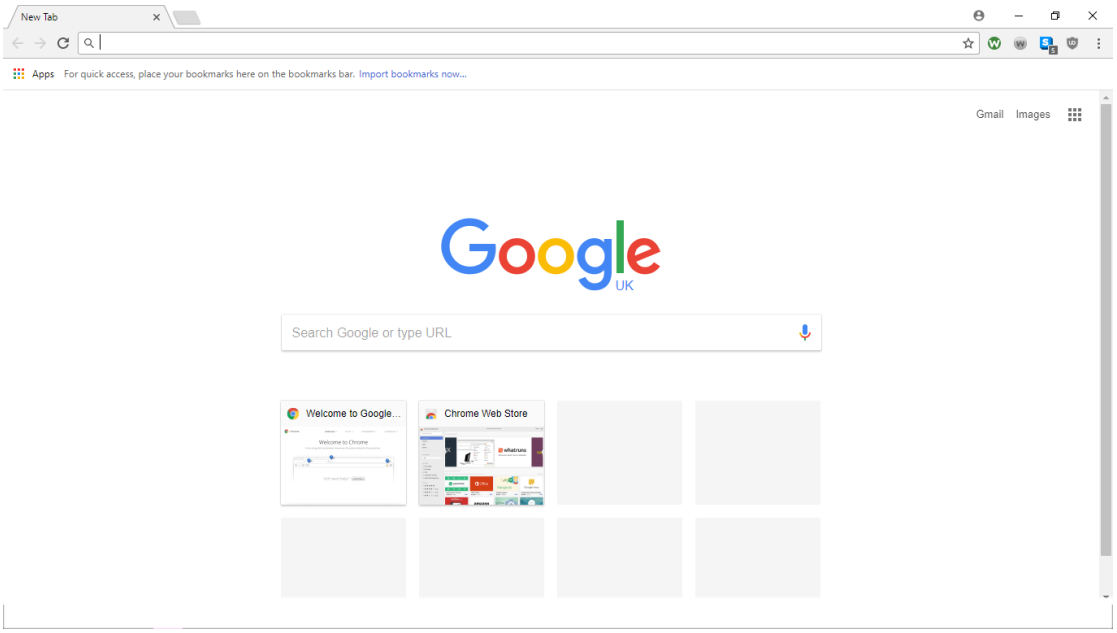


FIGURE 2: Google Homepage running on Google Chrome

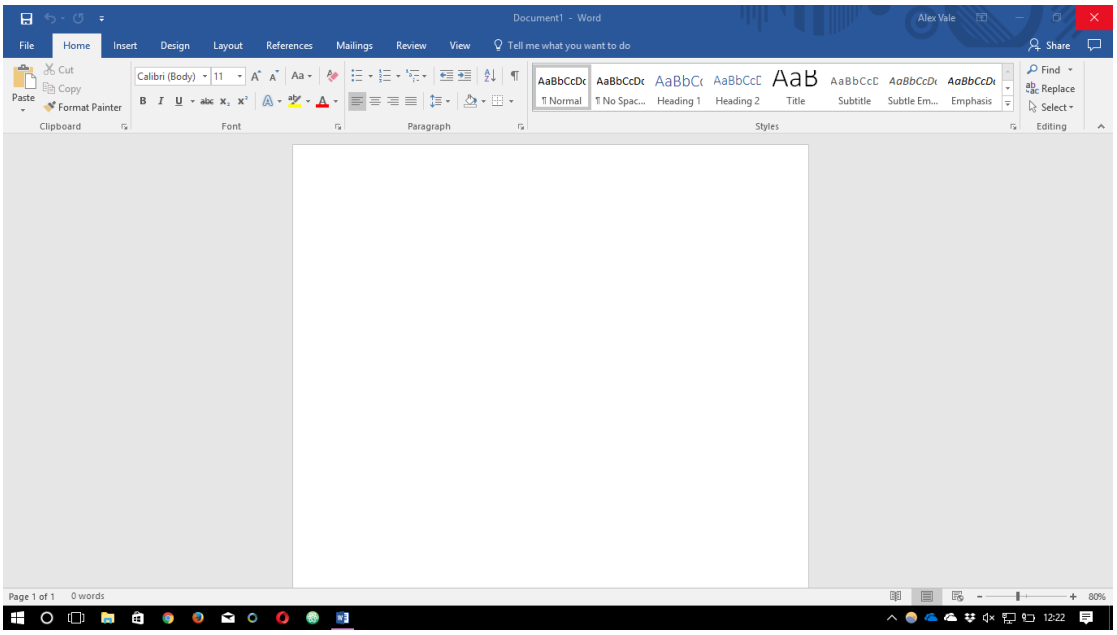


FIGURE 3: Word Blank Document

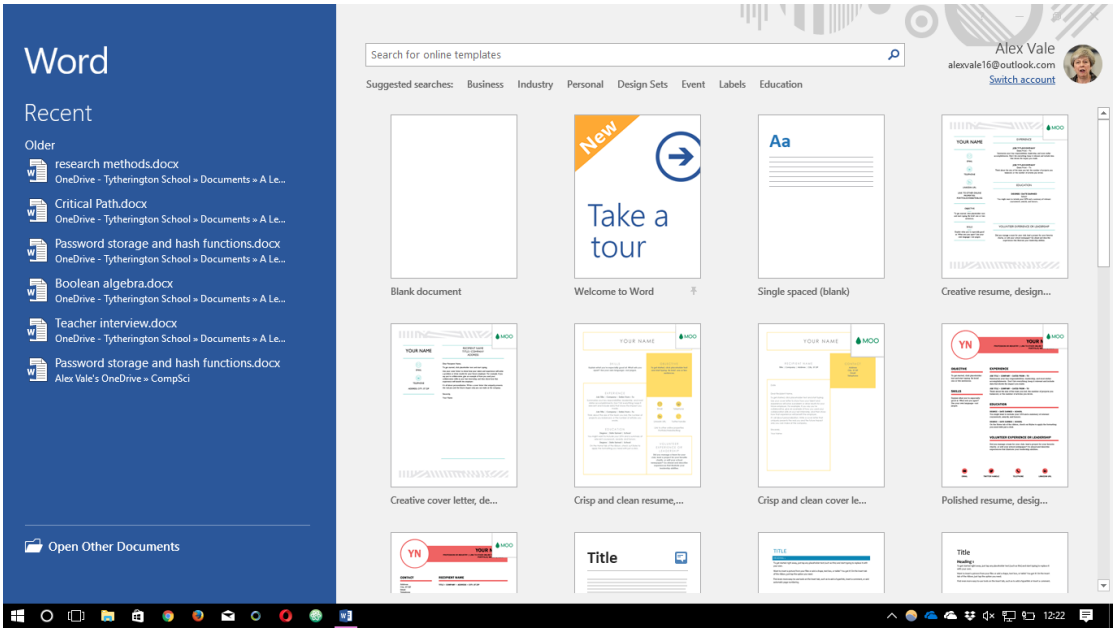


FIGURE 4: Word New Document

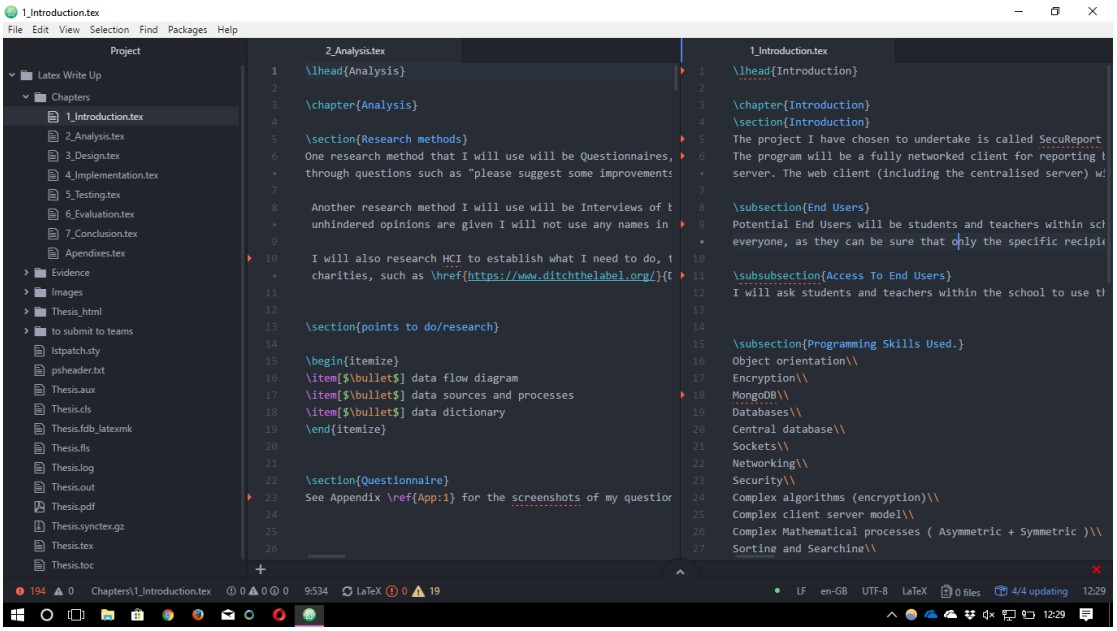


FIGURE 5: Atom Editor

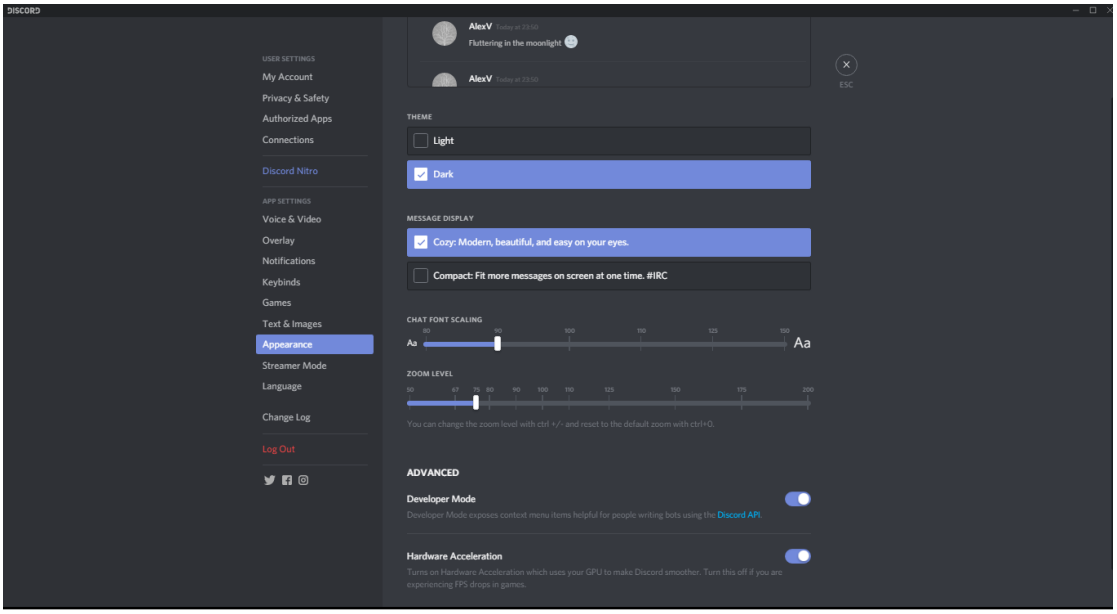


FIGURE 6: Discord Settings

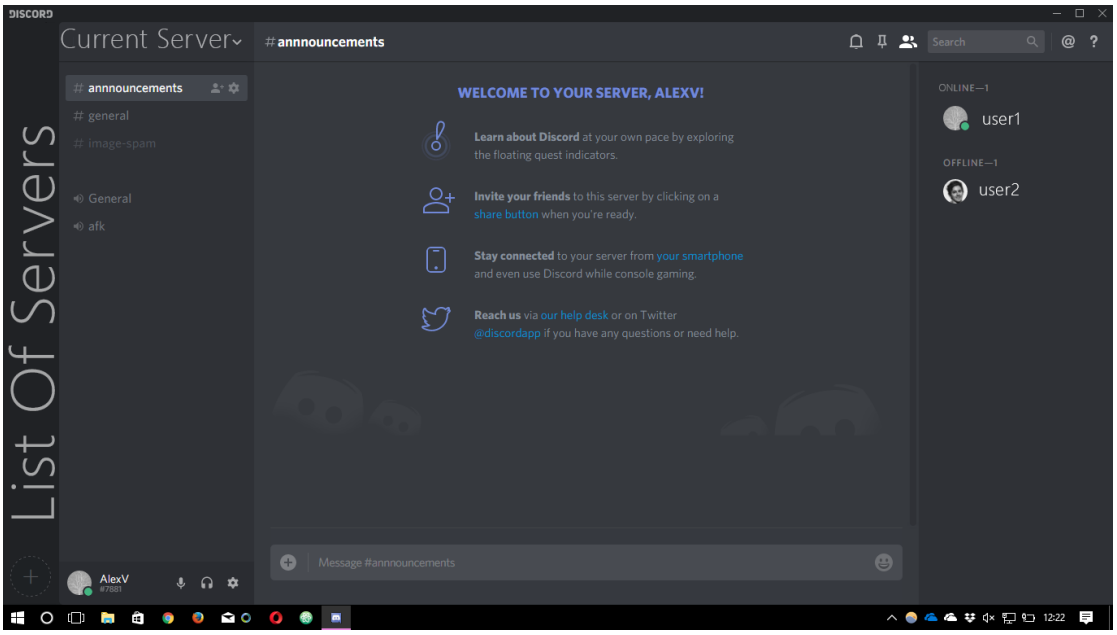


FIGURE 7: Discord Modified Server View

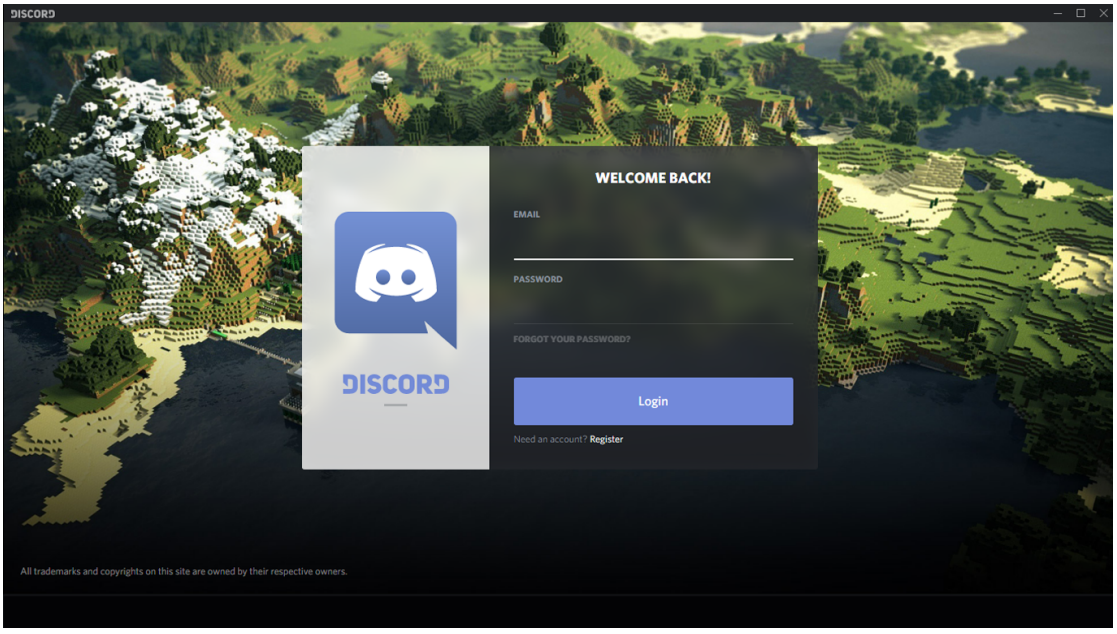


FIGURE 8: Discord Sign In

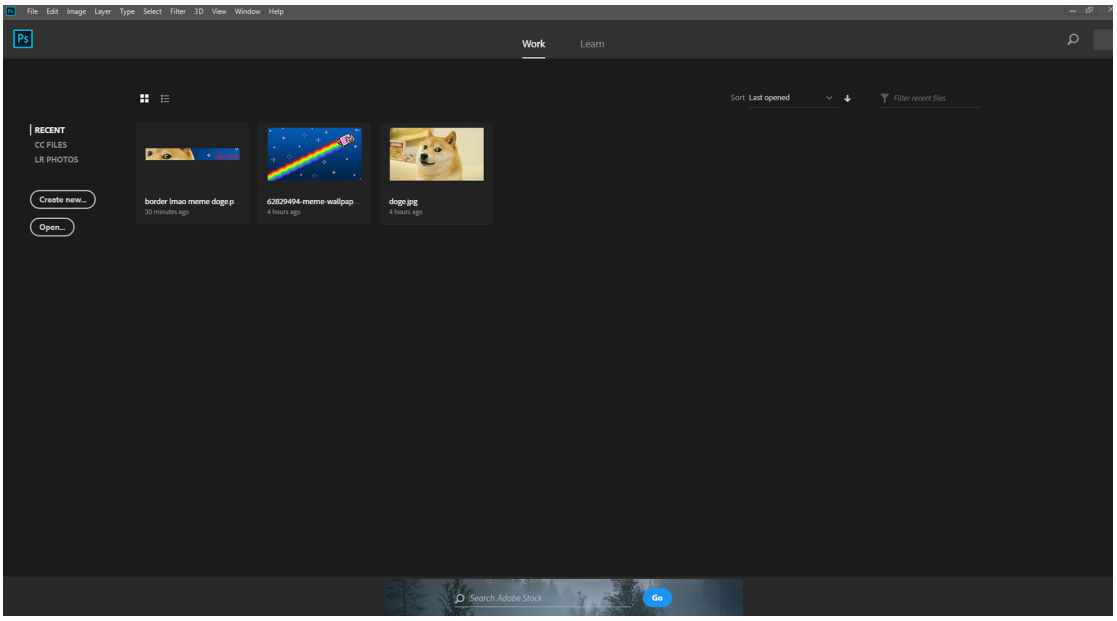


FIGURE 9: Photoshop

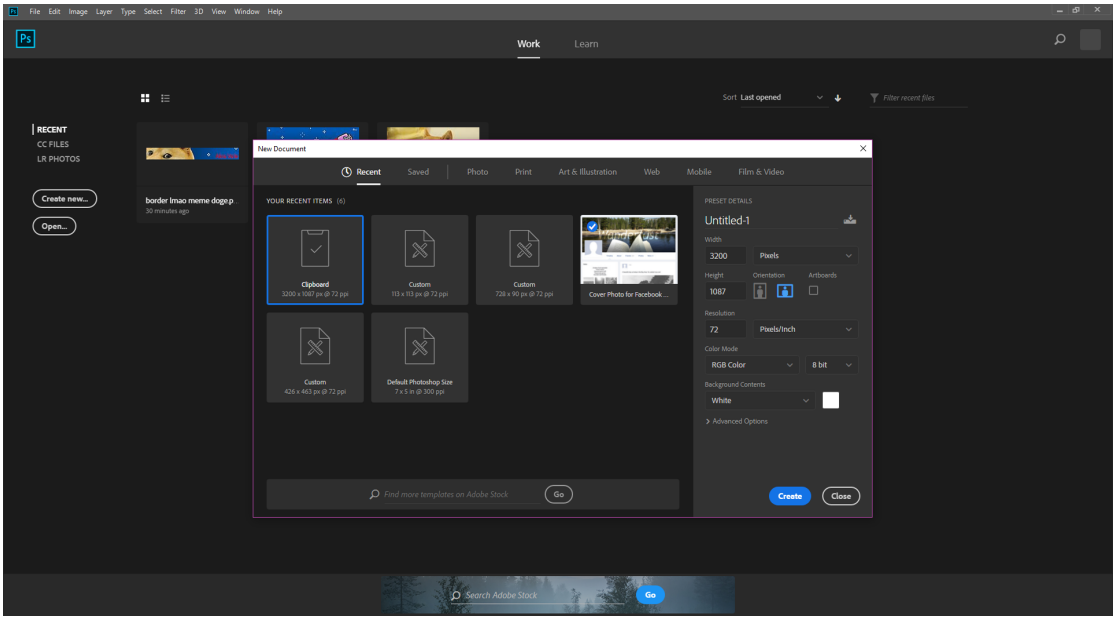


FIGURE 10: Photoshop New Document

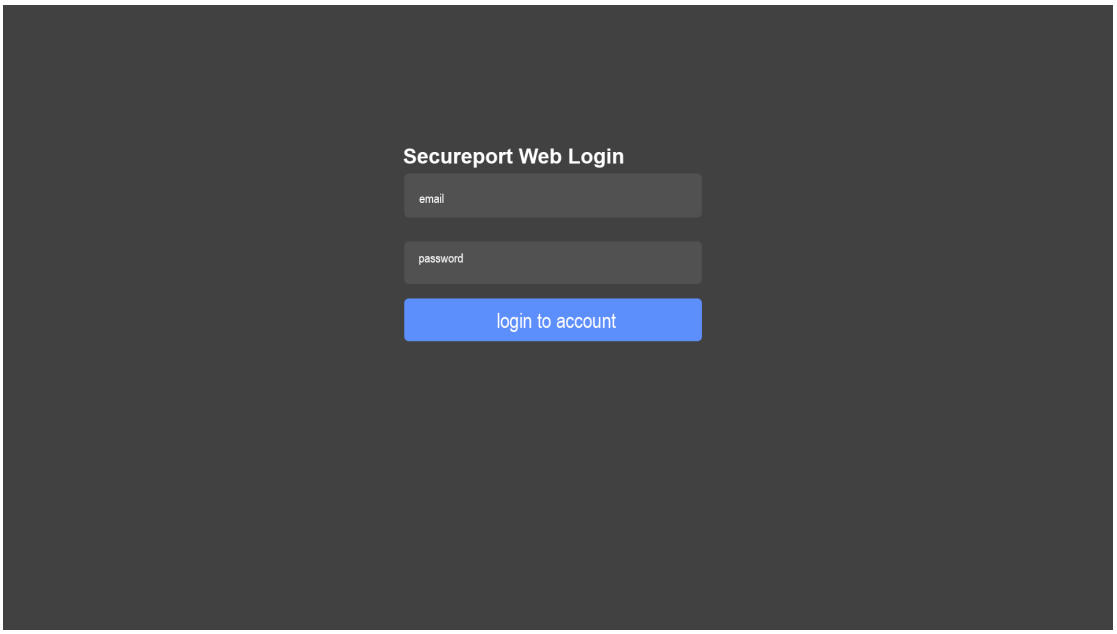


FIGURE 11: design mockup of application

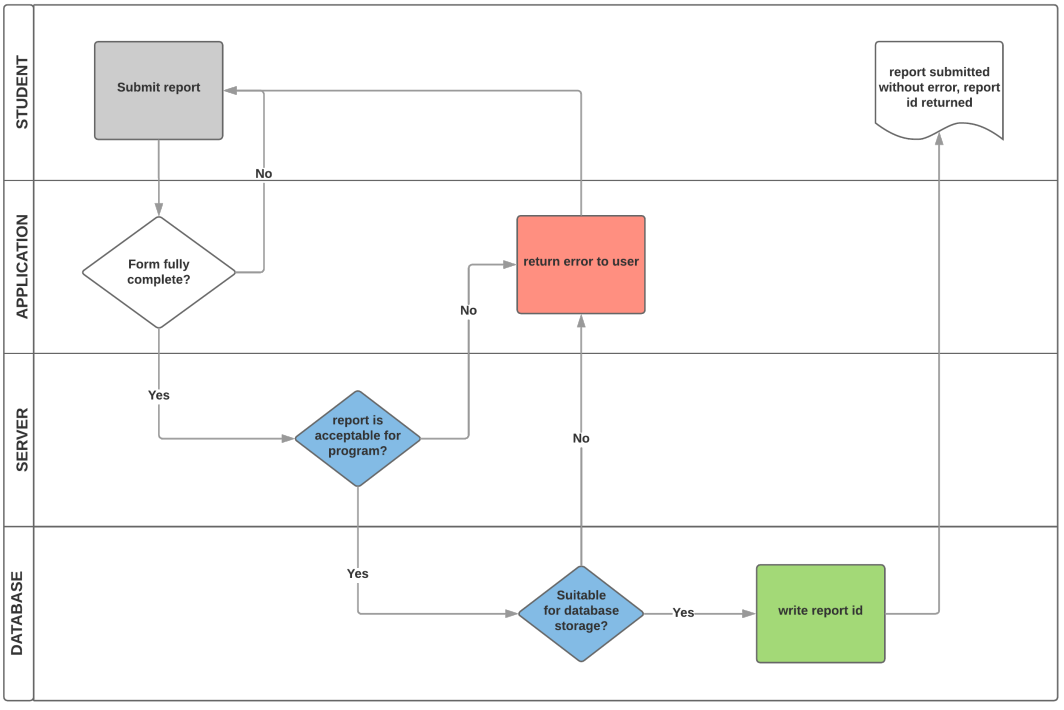


FIGURE 12: flowchart of report submission

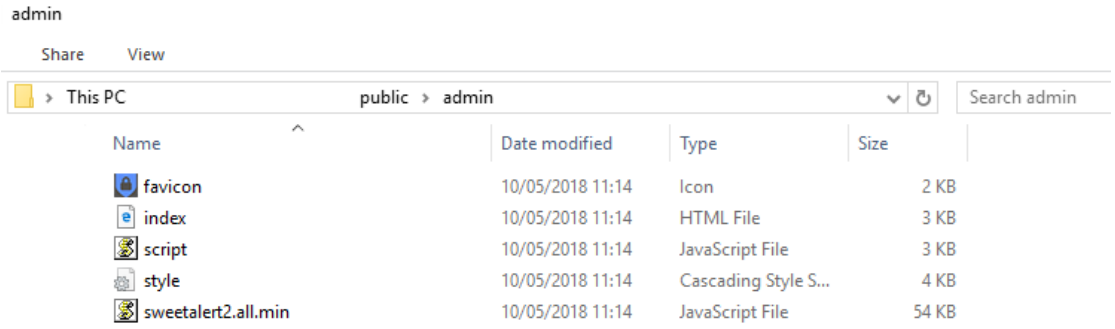


FIGURE 13: admin directory

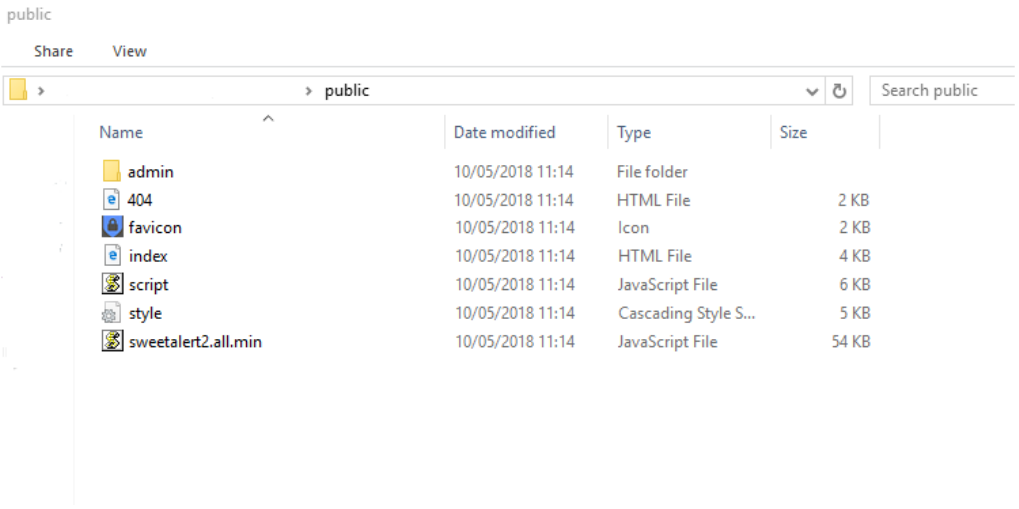


FIGURE 14: public directory

Public

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport"
    content="width=device-width, initial-scale=1">
    <title>Page Not Found</title>

    <style media="screen">
      body { background: #ECEFF1; color: rgba(0,0,0,0.87);
```

```
    font-family: Roboto,
    Helvetica, Arial, sans-serif; margin: 0; padding: 0; }
    #message { background: white; max-width: 360px; margin:
    100px auto 16px;
    padding: 32px 24px 16px; border-radius: 3px; }
    #message h3 { color: #888; font-weight: normal;
    font-size: 16px; margin:
    16px 0 12px; }
    #message h2 { color: #ffa100; font-weight: bold;
    font-size: 16px; margin:
    0 0 8px; }
    #message h1 { font-size: 22px; font-weight: 300;
    color: rgba(0,0,0,0.6);
    margin: 0 0 16px;}
    #message p { line-height: 140\%; margin: 16px 0 24px;
    font-size: 14px; }
    #message a { display: block; text-align: center;
    background: #039be5;
    text-transform: uppercase; text-decoration: none;
    color: white;
    padding: 16px; border-radius: 4px; }
    #message, #message a { box-shadow: 0 1px
    3px rgba(0,0,0,0.12),
    0 1px 2px rgba(0,0,0,0.24); }
    #load { color: rgba(0,0,0,0.4); text-align:
    center; font-size: 13px; }
    @media (max-width: 600px) {
      body, #message { margin-top: 0; background:
      white; box-shadow: none; }
      body { border-top: 16px solid #ffa100; }
    }
  </style>
</head>
<body>
  <div id="message">
    <h2>404</h2>
```

```
      <h1>Page Not Found</h1>
      <p>The specified file was not found on this website.
      Please check the URL for mistakes and try again.</p>
      <h3>Why am I seeing this?</h3>
      <p>This page was generated by the Firebase Command-Line Interface.
      To modify it, edit the <code>404.html</code> file in your project's
      configured <code>public</code> directory.</p>
    </div>
  </body>
</html>
```

```
    <html>
<head>
    <meta charset="UTF-8">
    <title>Secureport Student</title>

    <link href="https://fonts.googleapis.com/css?family=
    Nunito:400,600,700" rel="stylesheet">
    <!--
    <link rel="stylesheet" href="https://maxcdn.
    bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css">-->
    <link rel="stylesheet" href="./style.css">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery
    /3.3.1/jquery.min.js"></script>
    <script src="https://www.gstatic.com/firebasejs/4.12.1/
    firebase.js"></script>
    <script src="https://www.gstatic.com/firebasejs/4.12.1/
    firebase-firestore.js"></script>

    <script>
        // Initialize Firebase
        var config = {
            apiKey: "AIzaSyD1a920sL_Dzd1herQzUfnaeR5NIIsOSzM",
            authDomain: "port-a6eda.firebaseapp.com",
```

```
            databaseURL: "https://port-a6eda.firebaseio.com",
            projectId: "port-a6eda",
            storageBucket: "port-a6eda.appspot.com",
            messagingSenderId: "515080856207"
        };
        firebase.initializeApp(config);
    </script>



</head>
<body>

  <div id="login_div" class="main-div">
    <h3>SecuReport Web Login</h3>
    <input type="email" placeholder="Email..." id="email_field"/>
    <input type="password" placeholder="Password..." id="password_field" />

    <button onclick="login()">Login to Account</button>




  </div>
    <div id="anon_login-div" class="anon_main-div">
        <button onclick="login_anon()" >Submit report Anonymously</button>

   </div>

  <div id = "anon_user_div" class="loggedin-div">
      <h3>Welcome User</h3>
      <p>Welcome to a test version of secureport. You're currently
      logged in anonymously.</p>
      <p> This means that unfortunately you cannot view previous
      reports, but you can submit them regardless</p>
        <button onclick="logout()">Return to login</button>
```

```
  </div>

  <div id="user_div" class="loggedin-div">
    <h3>Welcome User</h3>
    <p id="user_para">Welcome to a test version of secureport.
    You're currently logged in.</p>
    <button onclick="logout()">Logout</button>
  </div>
<div class="container">
<div id="student-report" class="report-form student-form">

    <form id="newActivity">
    <header>
        <h3>Submit a new Report</h3>
        <div>Tell us what happened!</div>
        <p></p>
    </header>
    <div>
        <label class="desc" id="title1" for="Field1">Title: </label>
        <div>
            <input id="activityTitle" name="Field1" type="text"
            value="" tabindex="1" placeholder="brief summary of
            issue" required>
        </div>
    </div>
    <div>
        <label class="desc" id="description1" for="Field2">
        Description: </label>
        <div>
            <textarea id="activityDescription" name="Field2"
            spellcheck="true" rows="10" col="50" tabindex="2"
            placeholder="main body text" required></textarea>
        </div>
    </div>

    <div>
```

```
        <button id="saveForm" name="saveForm" type="submit">
        Submit Report</button>
    </div>

</form>
</div>

</div>

<div id="previous-reports" class="student-view">


</div>

</body>
<footer>


      <div id = "get-help" class="fragment">
          <span id='close'>X</span>
        <h3>Need help sooner?</h3>
        <p>There are many websites which can offer help immediately
        so you can feel happier in yourself and put the bullying
        behind you.</p>
        <a href="https://www.ditchthelabel.org/cyber-
        bullying-top-9-tips-on-overcoming-it/"> Cyber bullying tips </a>
        <br>
        <a href="https://www.childline.org.uk/info-advice/
        bullying-abuse-safety/types-bullying/">
        Types of bullying</a>
        <br>
        <a href="https://www.ditchthelabel.org/bullying-101/
        "> Bullying 101</a>
        <p>The National Bullying Helpline |
        CALL : 0845 22 55 787 | EMAIL : admin@national
        bullyinghelpline.co.uk</p>
```

```
      </div>

</footer>
<script src="script.js"></script>
<script src="sweetalert2.all.min.js"></script>
</html>



var anonuser = false;
var loggedin = false;
var show = "block";
var hide = "none";
var db = firebase.firestore();
var uid;

firebase.auth().onAuthStateChanged(function(user) {
  if (user) {
    // User is signed in.
    loggedin = true;

    document.getElementById("user_div").style.display = show;
    document.getElementById("login_div").style.display = hide;
    document.getElementById("anon_login-div").style.display = hide;
    document.getElementById("student-report").style.display= show;
    document.getElementById("previous-reports").style.display= show;

    \$(document.getElementsByClassName("fragment")).hide();

    var user = firebase.auth().currentUser;
      var email = user.email;
    uid = user.uid;
    console.log(email);
    console.log(uid);
```

```
      if  (email == null) {

          anonuser = true;
          document.getElementById("anon_user_div").style.display= show;
        document.getElementById("previous-reports").style.display= hide;
          document.getElementById("user_div").style.display = hide;

          // user is anonymous
          // save authData.uid to some local session info,
          to keep track of data

      } else {
          anonuser = false;
          document.getElementById("anon_user_div").style.display= hide;
      // user is logged in
      // transfer any data that you had stored from the
      // anonymous session to this new authUser.uid
  }
    if(user != null){

      var email_id = user.email;
      document.getElementById("user_para").innerHTML =
      "Welcome : "+ email_id;

    }

  } else {
    // No user is signed in.
    loggedin = false;
    document.getElementById("user_div").style.display = hide;
    document.getElementById("login_div").style.display = show;
    document.getElementById("anon_login-div").style.display = show;
    document.getElementById("student-report").style.display= hide;
    document.getElementById("previous-reports").style.display= hide;
    document.getElementById("anon_user_div").style.display= hide;
```

```
}
});
function login_anon(){
    firebase.auth().signInAnonymously().catch(function(error) {
  // Handle Errors here.
  var errorCode = error.code;
  console.log(errorCode);
  var errorMessage = error.message;
  console.log(errorMessage)

  // ...
});
};

function login(){

  var userEmail = document.getElementById("email_field").value;
  var userPass = document.getElementById("password_field").value;

  firebase.auth().signInWithEmailAndPassword(userEmail, userPass).
           catch(function(error) {
    // Handle Errors here.
    var errorCode = error.code;
    var errorMessage = error.message;
    swal({
        title: 'Error!',
        text: errorMessage,
        type: 'error',
        confirmButtonText: 'Dont worry I\'ll sort it now xoxo'

    })
    // ...
  });
```

```
}

function logout(){
  firebase.auth().signOut();
  loggedin = false;
}


\$('#newActivity').submit(function(event) {
    var \$form = \$(this);
    console.log("submit to Firebase");


    //make the submit disabled
    \$form.find("#saveForm").prop('disabled', false);

    //get the actual values that we will send to firebase
    var titleToSend = \$('#activityTitle').val();

    console.log(titleToSend);

    var descriptionToSend = \$('#activityDescription').val();

    console.log(descriptionToSend);


    //take the values from the form, and put them in an object
    var date = new Date();
    var unixtime = date.getTime();
    console.log(unixtime);

    console.log(uid);
    db.collection('test').doc(String(unixtime)).set({
        "description": descriptionToSend,
        "title": titleToSend,
        "status": "001",
```

```
        "resolution": "",
        "user": uid
    })
        .then(function (docRef) {
            console.log("Document written with ID: ", docRef.id);
        })
        .catch(function (error) {
            console.error("Error adding document: ", error);
        });

    return false;
});

\$(document).ready(function(){
    \$(document.getElementById("close")).click(function(){
        \$(document.getElementsByClassName("fragment")).fadeOut();
        document.getElementById("login_div").style.marginTop = "200px";
    });
});


\$("#previous-reports").height(\$("#student-report").height());

function addTable() {

    var myTableDiv = document.getElementById("previous-reports");
    var table = document.createElement('TABLE');
    var tableBody = document.createElement('TBODY');

    table.border = '1';
    table.appendChild(tableBody);

    var heading = new Array();
    heading[0] = "Title";
    heading[1] = "description";
    heading[2] = "resolution";
```

```
var report = new Array()

db.collection("test").get().then(function (querySnapshot) {
    var x = 0;
    querySnapshot.forEach(function (doc) {
        doc.data().user, uid);
        if (doc.data().user == uid) {
            console.log(doc.id, " => ", doc.data());
            report[x] = new Array(doc.data().title,
            doc.data().description, doc.data().resolution);
            x += 1;
        }

        else {
            x = x;
        }

    });
})
    .catch(function (error) {
        console.log("Error getting documents: ", error);
    });

//TABLE COLUMNS
var tr = document.createElement('TR');
tableBody.appendChild(tr);
for (i = 0; I < 3; i++) {
    var th = document.createElement('TH');
    th.width = '125';
    th.appendChild(document.createTextNode(heading[i]));
    tr.appendChild(th);
}

//TABLE ROWS
console.log("report length: ",report.length);
```

```
    for (i = 0; I < report.length; i++) {
        var tr = document.createElement('TR');
        tableBody.appendChild(tr);
        for (j = 0; j < 3; j++) {
            var td = document.createElement('TD');
            td.width = '125';
            console.log(report[i][j]);
            td.appendChild(document.createTextNode(report[i][j]));
            tr.appendChild(td)
        }
        tableBody.appendChild(tr);
    }
    myTableDiv.appendChild(table);
}

addTable();




    *{
overflow: hidden;
}

body {
  background: #515151;
  padding: 0px;
  margin: 0px;
  font-family: 'Nunito', sans-serif;
  font-size: 16px;
color: #fff;
}

input, button {
  font-family: 'Nunito', sans-serif;
  font-weight: 700;
}
```

```
.main-div, .loggedin-div {
  width: 65\%;
  margin: 0px auto;
  margin-top: 105px;
  padding: 0px;
  display: block;
  max-width: 350px;
  transition: all 2s;
  -webkit-transition: all 2s;
}
.anon_main-div{
  width: 65\%;
  margin: 0px auto;
  padding: 0px;
  display: block;
  max-width: 350px;
}

.main-div input,.main-div button,.anon_main-div button,
    .loggedin-div button{
  display: block;
  border: 1px solid #ccc;
  border-radius: px;
  background: #616161;
  padding: 15px;
  outline: none;
  width: 100\%;
  color: #FFFFFF;
  margin-bottom: 10px;
  transition: 0.3s;
  -webkit-transition: 0.3s;
  -moz-transition: 0.3s;
}
.loggedin-div button{
  width: 40\%;
```

```
}


.main-div button{
  border: 1.5px solid #5d8ffc;
}
.anon_main-div button{
border: 1.5px solid #FFB200;
}
.main-div input:focus {
  border: 1px solid #777;
}
#brief-advice{
  padding: 0px;
  margin: 0px;
  font-family: 'Nunito', sans-serif;
  font-size: 12px;
  color: #fff;
  position: absolute;
  width: 50\%;
  height: 150px;
  z-index: 15;
  top: 50\%;
  left: 5\%;
  margin-top: 200px;
}


.main-div button:hover, .loggedin-div button:hover {
  background: #515151;
  color: #5d8ffc;
  border: 1.5px solid #FFB200;
  cursor: pointer;
}
```

```
.anon_main-div button:hover{
  background: #515151;
  color: #FFB200;
  border: 1.5px solid #5d8ffc;
  cursor: pointer;
}

.swal2-popupl{
background: #515151 !important;
color: #FFFFFF;
}

.fragment {
    background: #616161;
    height: 23\%;
    border: 1px solid #ccc;
    border-radius: 3px;
    display: block;
    padding: 10px;
    padding-top: 5px;
    padding-right: 10px;
    box-sizing: border-box;
    text-decoration: none;
    margin: 10px;
    margin-top: 15\%;

}

.fragment:hover {
    box-shadow: 2px 2px 5px rgba(0,0,0,.2);

}

#close {
    float:right;
    display:inline-block;
```

```
    padding:0.2\% 0.5\%;
    color: #000;
    background:#ccc;
}

#close:hover {
        background:#ccc;
    color:#fff;
    }



*{
  padding: 2px;
}body {
  background: #515151;
  padding: 0px;
  margin: 0px;
  font-family: 'Nunito', sans-serif;
  font-size: 16px;
color: #fff;
}

input, button {
  font-family: 'Nunito', sans-serif;
  font-weight: 700;
}
textarea{
  width: 40\%;
  resize: none;

}
.loggedin-div {
    top: -10\%;
    display: none;
    width: 45\%;
    margin-left: 7\%;
```

```
}

.report-form {
  position: fixed;
  display: none;
  width: 50\%;
  margin-left: -15\%;
  margin-top: 50px;
  margin-right: 200px;
  padding: 10px;
  max-width: 600px;
}
.report-form input,#activityDescription, .report-form button{
  top:150px;
  display: block;
  border: 1px solid #ccc;
  border-radius: 2px;
  background: #616161;
  padding: 15px;
  outline: none;
  width: 100\%;
  color: #FFFFFF;
  margin-bottom: 20px;
  transition: 0.3s;
  -webkit-transition: 0.3s;
  -moz-transition: 0.3s;
}

.report-form input:focus {
  border: 1px solid #777;
}

./report-form button, ./loggedin-div button{

  margin:  auto;
  background: #5d8ffc;
```

```
  color: #515151;
  border: 1.5px solid #FFB200;
  border-radius: 5px;
  padding: 15px;
  width: 60\%;
  transition: 0.3s;
  -webkit-transition: 0.3s;
  -moz-transition: 0.3s;
}

.report-form button:hover, .loggedin-div button:hover {
  background: #515151;
  color: #5d8ffc;
  border: 1.5px solid #FFB200;
  cursor: pointer;
}

.anon_report-form button{
  background: #FFB200;
  color: #515151;
  border: 1.5px solid #5d8ffc;
  border-radius: 5px;
  padding: 15px;
  display: block;
  width: 65\%;
  transition: 0.3s;
  -webkit-transition: 0.3s;
  -moz-transition: 0.3s;
}

.anon_report-form button:hover{
  background: #515151;
  color: #FFB200;
  border: 1.5px solid #5d8ffc;
  cursor: pointer;
}
```

```
.swal2-popupl{
background: #515151 !important;
color: #FFFFFF;
}
#anon_user_div{
  width:50\%;
  position:fixed;
  top:100px;
  left:60\%;
  max-height: 480px;
}
#anon_user_div button{
  width: 100\%

}
#previous-reports {
    position:fixed;
    top:360px;
    left:60\%;
    max-height: 480px;
}
#previous-reports{
    align-items: center;
    padding: 5px;
        background-color: #515151;
        align-content: center;
    }
#previous-reports tr {
    height: 45px;
    padding: 20px;
    border-bottom: 1px solid #ccc;
}
#previous-reports td{
    width: 200px;
```

```
}
```

Admin

```
    <html>
<head>
    <meta charset="UTF-8">
    <title>Secureport Admin</title>
    <link href="https://fonts.googleapis.com/css?
    family=Nunito:400,600,700" rel="stylesheet">
    <script src="https://ajax.googleapis.com/ajax/
    libs/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://www.gstatic.com/firebasejs
    /4.12.1/firebase.js"></script>
    <script src="https://www.gstatic.com/firebasejs
    /4.12.1/firebase-firestore.js"></script>
    <script>
        // Initialize Firebase
        var config = {
            apiKey: "AIzaSyD1a920sL_Dzd1herQzUfnaeR5NIIsOSzM",
            authDomain: "port-a6eda.firebaseapp.com",
            databaseURL: "https://port-a6eda.firebaseio.com",
            projectId: "port-a6eda",
            storageBucket: "port-a6eda.appspot.com",
            messagingSenderId: "515080856207"
        };
        firebase.initializeApp(config);
    </script>
    <link rel="stylesheet" href="./style.css" />
</head>
<body>

  <div id="login_div" class="main-div">
    <h3>SecuReport Web Login</h3>
    <input type="email" placeholder="Email..."
```

```
    id="email_field"/>
    <input type="password" placeholder="Password..."
    id="password_field" />

    <button onclick="login()">Login to Account</button>




  </div>
    <div id="anon_login-div" class="anon_main-div">
        <button onclick="login_anon()" >Submit report
        Anonymously</button>

   </div>
  <div id="user_div" class="loggedin-div">
    <h3>Welcome User</h3>
    <p id="user_para">Welcome to a test version of
    secureport.You're currently logged in.</p>
    <button onclick="logout()">Logout</button>
  </div>


  <div id = admin-data>
      <p id="data"></p>
  </div>



</body>
<footer>


</footer>

    <script src="script.js"></script>
```

```
    <script src="sweetalert2.all.min.js"></script>
</html>



var db = firebase.firestore();

firebase.auth().onAuthStateChanged(function(user) {
  if (user) {
    // User is signed in.

    document.getElementById("user_div").style.display = "block";
    document.getElementById("login_div").style.display = "none";
    document.getElementById("anon_login-div").style.display = "none";
    document.getElementById("admin-data").style.display= "block";
    \$(document.getElementsByClassName("fragment")).hide();

    var user = firebase.auth().currentUser;

    if(user != null){

      var email_id = user.email;
      document.getElementById("user_para").innerHTML =
      "Welcome : " + email_id;

    }

  } else {
    // No user is signed in.

    document.getElementById("user_div").style.display = "none";
    document.getElementById("login_div").style.display = "block";
    document.getElementById("anon_login-div").style.display = "block";
    document.getElementById("admin-data").style.display= "none";
```

```
}
});
function login_anon(){
    firebase.auth().signInAnonymously().catch(function(error) {
  // Handle Errors here.
  var errorCode = error.code;
  var errorMessage = error.message;
  // ...
});
}
function login(){

  var userEmail = document.getElementById("email_field").value;
  var userPass = document.getElementById("password_field").value;

  firebase.auth().signInWithEmailAndPassword
  (userEmail, userPass).catch(function(error) {
    // Handle Errors here.
    var errorCode = error.code;
    var errorMessage = error.message;
    swal({
        title: 'Error!',
        text: errorMessage,
        type: 'error',
        confirmButtonText: 'Dont worry I\'ll sort it now xoxo'
    })
    // ...
  });


}

function logout(){
  firebase.auth().signOut();
}
```

```
db.collection("test").where("status", "==", "001")
    .get()
    .then(function (querySnapshot) {
        querySnapshot.forEach(function (doc) {
           // doc.data() is never undefined for query doc snapshots
            console.log(doc.id, " => ", doc.data());
            document.getElementById("data").innerHTML = string(doc.id
            + " => ") + doc.data().title + " || " + doc.data().description;
        });
    })
    .catch(function (error) {
        console.log("Error getting documents: ", error);
    });


\$(document).ready(function(){
    \$(document.getElementById("close")).click(function(){
        \$(document.getElementsByClassName("fragment")).fadeOut();
        document.getElementById("login_div").style.marginTop = "200px";
    });
});




    *{
overflow: hidden;
}

body {
  background: #515151;
  padding: 0px;
  margin: 0px;
  font-family: 'Nunito', sans-serif;
  font-size: 16px;
color: #fff;
```

```
}

input, button {
  font-family: 'Nunito', sans-serif;
  font-weight: 700;
}

.main-div, .loggedin-div {
  width: 65\%;
  margin: 0px auto;
  margin-top: 105px;
  padding: 0px;
  display: block;
  max-width: 350px;
  transition: all 2s;
  -webkit-transition: all 2s;
}
.anon_main-div{
  width: 65\%;
  margin: 0px auto;
  padding: 0px;
  display: block;
  max-width: 350px;
}

.main-div input,.main-div button,.anon_main-div button, .loggedin-div button{
  display: block;
  border: 1px solid #ccc;
  border-radius: px;
  background: #616161;
  padding: 15px;
  outline: none;
  width: 100\%;
  color: #FFFFFF;
  margin-bottom: 20px;
  transition: 0.3s;
```

```
  -webkit-transition: 0.3s;
  -moz-transition: 0.3s;
}
.main-div button{
  border: 1.5px solid #5d8ffc;
}
.anon_main-div button{
border: 1.5px solid #FFB200;
}
.main-div input:focus {
  border: 1px solid #777;
}
#brief-advice{
  padding: 0px;
  margin: 0px;
  font-family: 'Nunito', sans-serif;
  font-size: 12px;
  color: #fff;
  position: absolute;
  width: 50\%;
  height: 150px;
  z-index: 15;
  top: 50\%;
  left: 5\%;
  margin-top: 200px;
}


.main-div button:hover, .loggedin-div button:hover {
  background: #515151;
  color: #5d8ffc;
  border: 1.5px solid #FFB200;
  cursor: pointer;
}
```

```
.anon_main-div button:hover{
  background: #515151;
  color: #FFB200;
  border: 1.5px solid #5d8ffc;
  cursor: pointer;
}

.swal2-popupl{
background: #515151 !important;
color: #FFFFFF;
}

.fragment {
    background: #616161;
    height: 23\%;
    border: 1px solid #ccc;
    border-radius: 3px;
    display: block;
    padding: 10px;
    padding-top: 5px;
    padding-right: 10px;
    box-sizing: border-box;
    text-decoration: none;
    margin: 10px;
    margin-top: 15\%;

}

.fragment:hover {
    box-shadow: 2px 2px 5px rgba(0,0,0,.2);

}

#close {
    float:right;
    display:inline-block;
```

```
    padding:0.2\% 0.5\%;
    color: #000;
    background:#ccc;
}

#close:hover {
        background:#ccc;
    color:#fff;
    }



*{
  padding: 2px;
}body {
  background: #515151;
  padding: 0px;
  margin: 0px;
  font-family: 'Nunito', sans-serif;
  font-size: 16px;
color: #fff;
}

input, button {
  font-family: 'Nunito', sans-serif;
  font-weight: 700;
}
textarea{
  width: 40\%;
  resize: none;

}
.report-form, .loggedin-div {
  display: none;
  width: 85\%;
  margin-left: 5\%;
  margin-top: 200px;
```

```
  margin-right: 200px;
  padding: 10px;
  max-width: 1600px;
}
.report-form input,#activityDescription, .report-form button{
  display: block;
  border: 1px solid #ccc;
  border-radius: 2px;
  background: #616161;
  padding: 15px;
  outline: none;
  width: 40\%;
  color: #FFFFFF;
  margin-bottom: 20px;
  transition: 0.3s;
  -webkit-transition: 0.3s;
  -moz-transition: 0.3s;
}

.report-form input:focus {
  border: 1px solid #777;
}

./report-form button, ./loggedin-div button{

  margin:  auto;
  background: #5d8ffc;
  color: #515151;
  border: 1.5px solid #FFB200;
  border-radius: 5px;
  padding: 15px;
  width: 40\%;
  transition: 0.3s;
  -webkit-transition: 0.3s;
  -moz-transition: 0.3s;
}
```

```
.report-form button:hover, .loggedin-div button:hover {
  background: #515151;
  color: #5d8ffc;
  border: 1.5px solid #FFB200;
  cursor: pointer;
}

.anon_report-form button{
  background: #FFB200;
  color: #515151;
  border: 1.5px solid #5d8ffc;
  border-radius: 5px;
  padding: 15px;
  display: block;
  width: 100\%;
  transition: 0.3s;
  -webkit-transition: 0.3s;
  -moz-transition: 0.3s;
}

.anon_report-form button:hover{
  background: #515151;
  color: #FFB200;
  border: 1.5px solid #5d8ffc;
  cursor: pointer;
}

.swal2-popupl{
background: #515151 !important;
color: #FFFFFF;
}
```

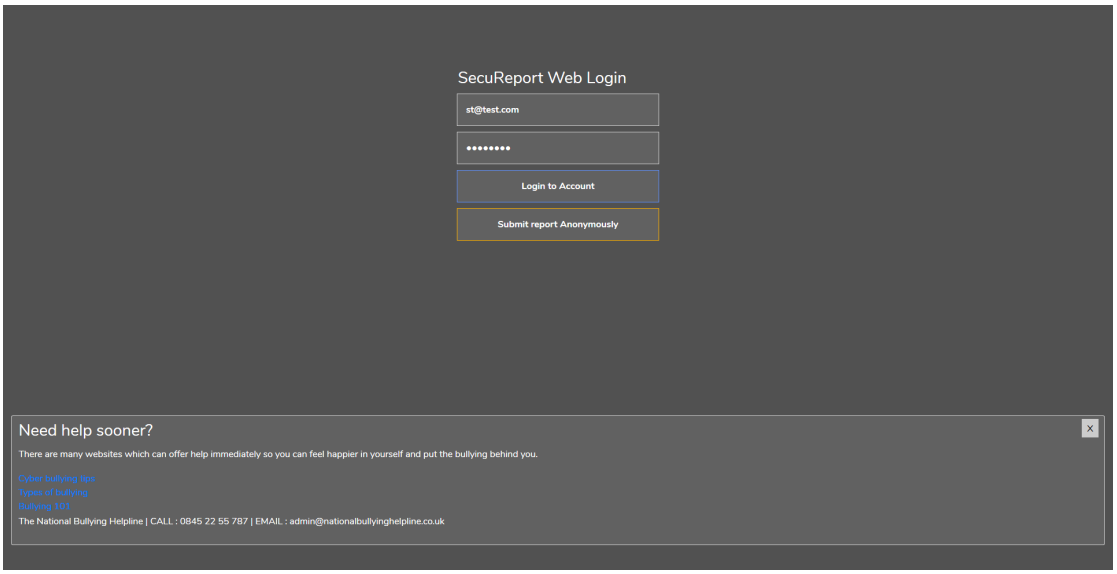


FIGURE 15: accessible

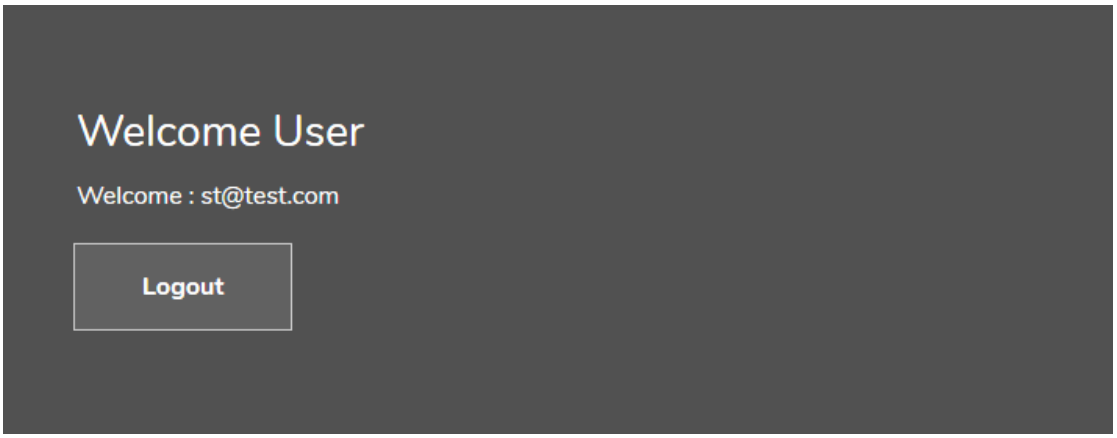


FIGURE 16: log in

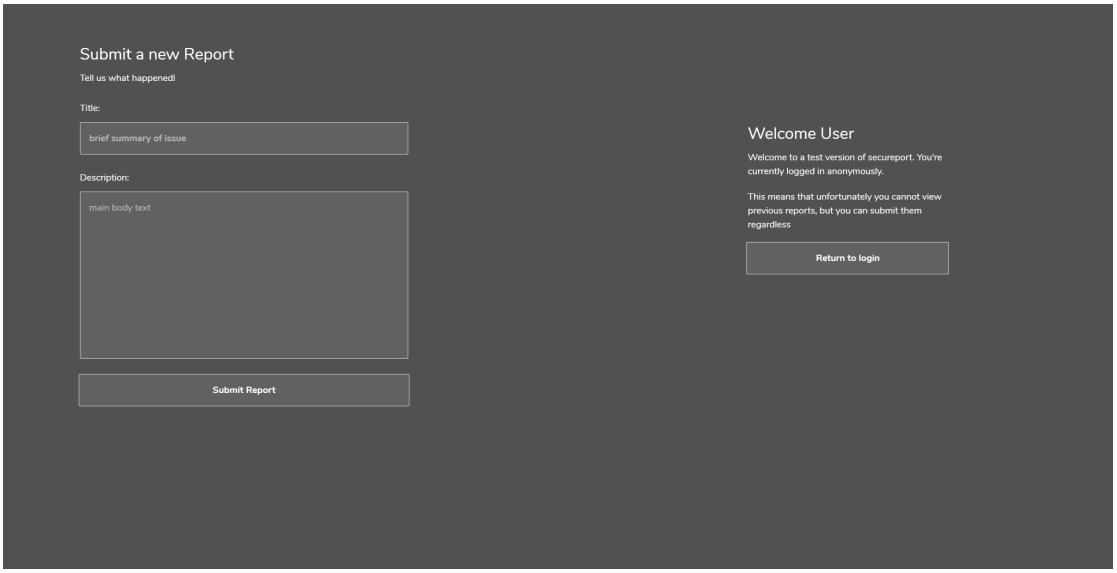


FIGURE 17: anonymous log in

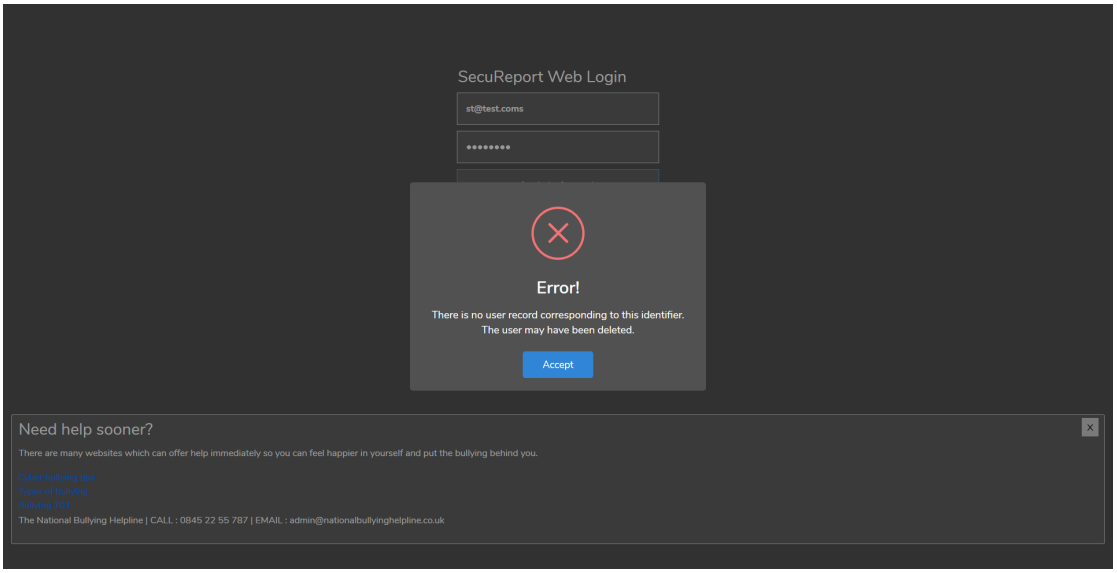


FIGURE 18: wrong email

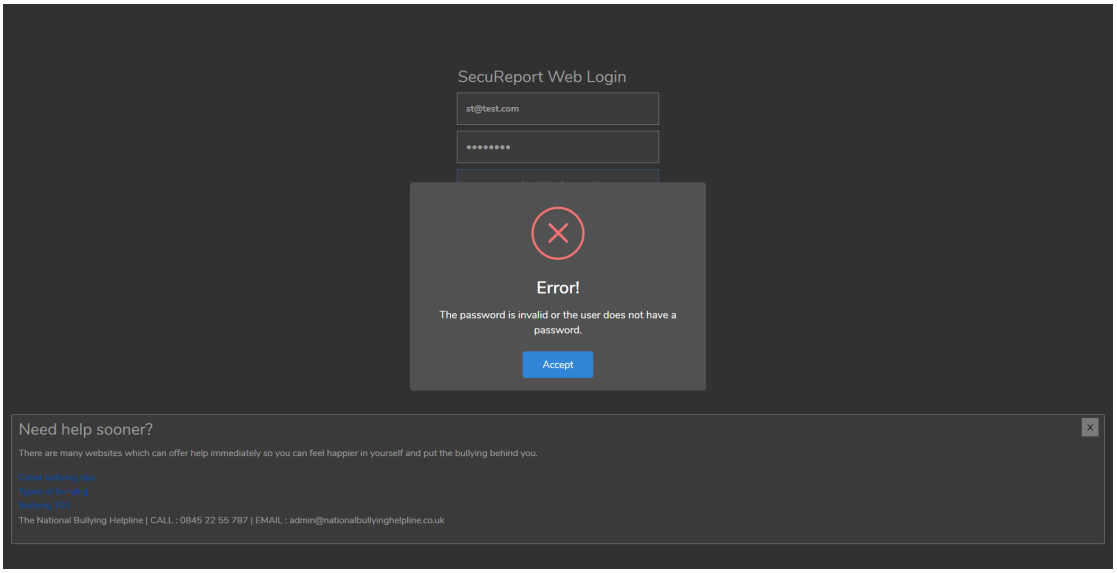


FIGURE 19: wrong password

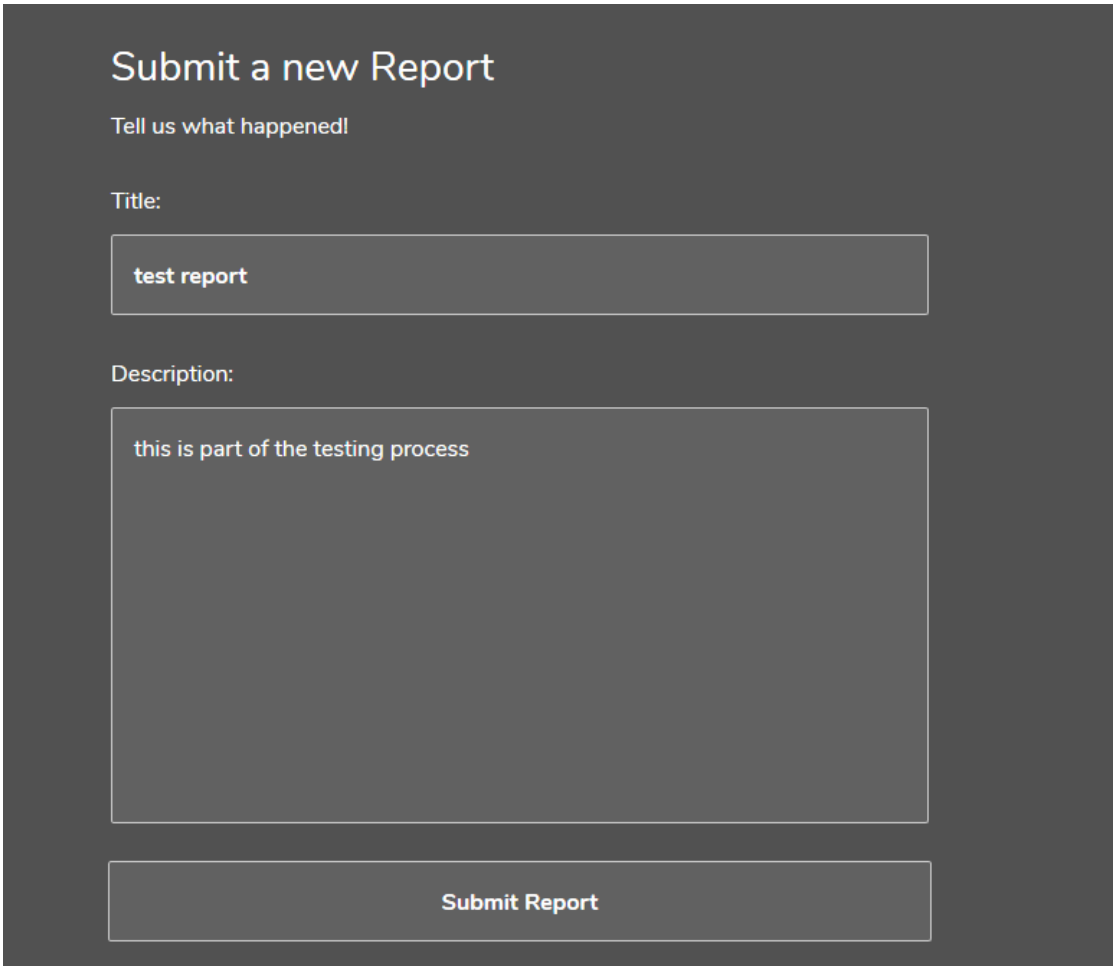


FIGURE 20: submit report

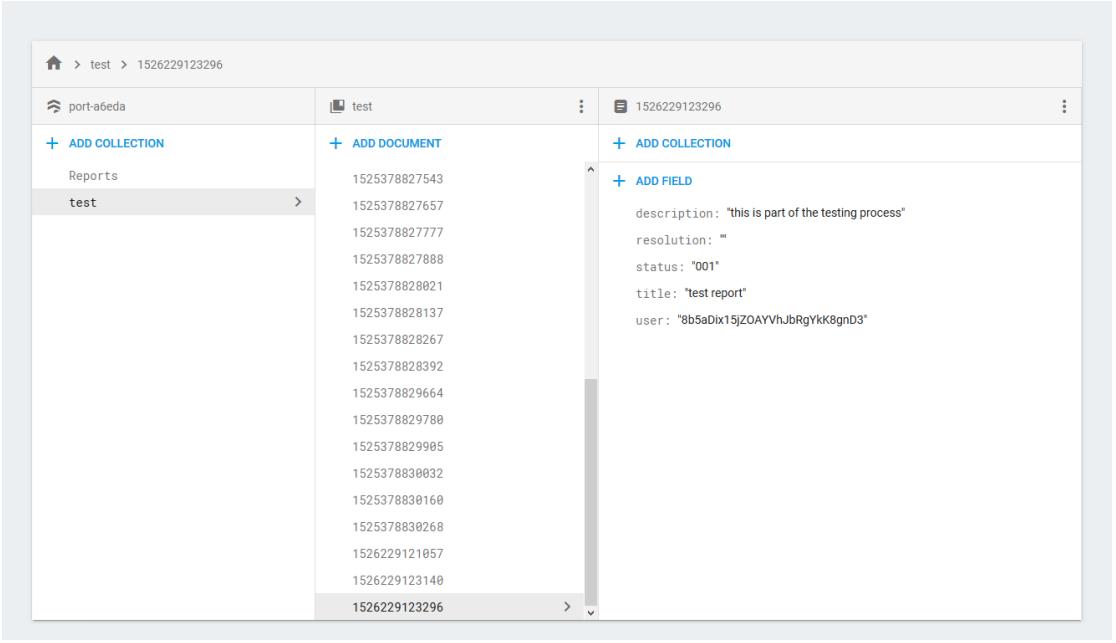


FIGURE 21: retrieve report