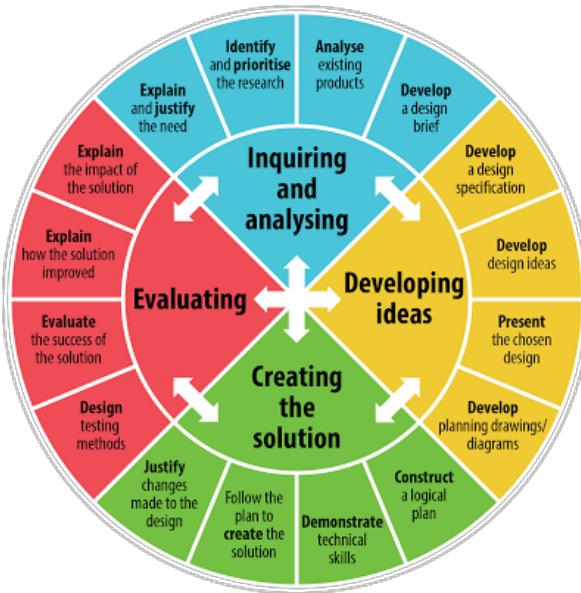


Digital Design

Help the Community

NAME of Student: Rohan Goyal

<p><u>Statement of Inquiry:</u></p> <p>The design and development of a product needs to consider the requirements of the intended user and the perspective customer while creating innovative interactions for the communities.</p>	<p>Global Context: Fairness and development- imagining a hopeful future.</p> <p>Key Concept: Communities</p> <p>Related Concept: Innovation/Perspective</p>			
<p>Design Situation:</p> <p>What does the word “community” mean to you? For some, community means the four-block radius surrounding their homes. For others, community means their entire neighborhood, city, or town. Regardless of how you define your community, you should know that you can play an active role in building it and helping it grow.</p> <p>But how? There are countless opportunities to get started, and here’s the best part: You don’t need to be rich, or powerful, or of a certain age to make a real impact. The way to really make a difference over time is a series of small steps.</p> <p>Supporting your local community can bring a whole host of benefits. Lending a helping hand can help you to make new friends, learn new skills, advance your career and, most importantly, give something back to your local community.</p>	<p>Inquiry Questions:</p> <p>Factual</p> <ul style="list-style-type: none">• Which technologies have helped you as a student community? <p>Conceptual</p> <ul style="list-style-type: none">• How to identify fake images?• How do fake images impact our perspectives? <p>Debatable</p> <ul style="list-style-type: none">• Can technology help solve every problem in the world?			
<p>Design Task:</p> <p>In this project, you will be required to help a client solve a real-world problem for a community.</p> <p>You must have a client for this project and a target community. You will need to collaborate with the client to get information, and your client will be part of the evaluation process to determine the success of your project. It could be a movie, product, prototype or a social media campaign.</p>	<p>Deliverables:</p> <ol style="list-style-type: none">1. Folder: Help the Community2. Files: Documents for each stage of the Design Cycle3. Academic Honesty Pledge4. Design Cycle and the cover page for each stage.5. All pictures/videos/other resources used organized properly			
<p>Criterion Assessed: A,B,C,D</p>				
<p>ATL Skills:</p> <ul style="list-style-type: none">• Communication Skills• Organisation Skills• Reflection Skills• Critical Thinking Skills	<table><tr><td>Collaboration skills</td></tr><tr><td>Affective Skills</td></tr><tr><td>Literacy Skills</td></tr></table>	Collaboration skills	Affective Skills	Literacy Skills
Collaboration skills				
Affective Skills				
Literacy Skills				



ACADEMIC HONESTY STATEMENT

My signature below constitutes my pledge that all of the writing is my own work, with the exception of those portions which are properly documented.

I understand and accept the following definition of plagiarism:

1. Plagiarism includes the literal repetition without acknowledgment of the writings of another author. All significant phrases, clauses, or passages in this paper which have been taken directly from source material have been enclosed in quotation marks and acknowledged in the text itself as well as in the list of Works Cited or Bibliography.
2. Plagiarism includes borrowing another's ideas and representing them as my own. To paraphrase the thoughts of another writer without acknowledgment is to plagiarize. Plagiarism also includes inadequate paraphrasing. Paraphrased passages (those put into my own words) have been properly acknowledged in the text and in the bibliography.
3. Plagiarism includes using another person or organization to prepare this paper and then submitting it as my own work.

I understand what plagiarism is, and I further understand that if plagiarism is found in my paper, my professor will follow the procedures on academic dishonesty set forth by Raffles World Academy.

Signed: Rohan Goyal

Date : 28/9/2018

(Adapted from Writing at Carson-Newman College, 4th Ed.)

Criterion A

Problem Explanation

Cyberbullying essentially refers to aggressive and bullying behaviour conducted over online channels, such as social media or chat rooms. Cyberbullying is such a problem because of how it is almost ubiquitous- the internet is infamous for its 'trolls' and... poor standard of discourse, to put it lightly. Since it occurs online, cyberbullying is essentially omnipresent Targeted bullying is also more common online, since social media offers a degree of anonymity and immunity from consequences- thus, it is a far less level playing field for potential bullies. Cyberbullying is most common among the community of teenagers/high students, since not only do they spend, but they are at the time when their self-esteem tends to be most fragile. Cyberbullying is thus not only prevalent among teens, but it is also the most damaging. Nearly 30% of children have reported incidents of cyberbullying on twitter alone, and 20% of children report that this makes them reluctant to go to school. Cyber-bulling can thus clearly impact victims' emotional and psychological functions, and cannot be overlooked.

The global context of this unit is fairness and development. Since my product focuses on a technological problem, and thus is technological in nature itself, it is closely related to development. It is also intended to improve the overall well-being of the community, through both technological and moral development. My project is also related to fairness as it is intended to combat bullying, which is essentially a form of power imbalance and unfairness. In addition, I have to make sure that the benefits of the product are equally and fairly shared among people. Ultimately, my project aims to counter an unfairness which is caused by technological development through its own use of technology, which fits in very well with the global context.

Client and Audience: My client is Ms Rekha Simon who is affiliated with the Anti Bullying Committee (henceforth ABC) of Raffles World Academy (henceforth RWA). The committee is dedicated to preventing bullying among students, and to creating awareness of the problem among students. Recently Ms Rekha discussed cyberbullying and potential solutions with us (which is in fact where I got the idea from). The main target audience for my product will be RWA secondary school students; however, I intend it to work for primary students equally well. To that end the app should require minimal effort and/or technical skill. Cyberbullying is a problem which is difficult to combat by conventional means, and a technological approach is necessary.

Brainstorming

Aspects of the problem:

- Difficult to avoid and disconnect (for victims)
- Bullies often don't face consequences, since it is anonymous
- Often kept secret/dismissed/unacknowledged
- Difficult to deal with, due to the administration required for investigating/dealing with complaints (which is ripe for automation).

Possible products

- A newsletter/blog/campaign against cyberbullying
- Online forum for people to talk about experiences of cyberbullying, and potentially get support
- An app or extension which lets students report instances of cyberbullying directly and automatically
- An app which creates 'friction' for cyberbullying such as censoring curse words or asking 'are you sure?' (c.f rethink).
- A movie or other presentation explaining the effects of cyberbullying, in order to discourage prospective bullies.

Final Solution and Justification

My final solution is to create a browser extension which allows students of RWA to report abusive social media posts to the ABC. This is because often the barrier to reporting abuse is that it takes time for the report to be acknowledged. If the incident can be reported directly to the ABC, that problem should be solved. This efficient method of detecting abuse also allows for making bullies face some consequences such as strongly worded emails (admittedly not much of a consequence, but it's a start), eliminating the defence of anonymity. This is also something which I think is focused on working *with* the community, rather than preaching to it such as with a campaign or a video. The app is thus user-focused, which is a key principle of design (especially when the focus is explicitly on clients and communities).

Problem Statement: How might we decrease prevalence of cyberbullying among RWA students through focused use of technology to increase awareness and accountability?

Analysis of products:

These products can only be used by a school, not downloaded by individuals. This limited my ability to analyse them, since I could not try them out for myself, only refer to websites and online reviews (which are cited in appendix A).

Product name	Good Features	Bad Features	Overall Evaluation
Speak Up	<p>One important feature is that the app allows for anonymous reporting, which is important since ‘snitching’ can lead to revenge from bullies.</p> <p>Has features such as statistical analysis for privileged users such as teachers and admins. This can help prevent bullying even before it happens.</p> <p>The app tackles all forms of bullying, both cyber and physical. It has both ‘choice’ options, for filling in the type of bullying, and general qualitative data for describing the incident, which helps for more nuanced cases.</p> <p>The app works in real time with no delay, which is a boon for adults/admins.</p> <p>There is a high level of adaptability, since almost none of the system is set in stone. The app essentially provides information, and is focused more on facilitating other activities, rather than on creating a solution entirely in itself.</p> <p>The UI of the app is simple to use. Since it has only a single function, the method for sending reports is minimalist and simple. Students need only select the type and location of the incident, type a brief description, and send it. The choice of anonymous reporting is also simple- just a single checkbox.</p>	<p>Does not require evidence, thus creating the possibility of spamming and false incidents.</p> <p>Requires the whole school to enrol to work; individual students are thus powerless.</p> <p>Only accessible on phones and tablets (not online), making it difficult for most schools to use (since phones are rarely allowed).</p> <p>A screenshot can often reveal the name/account of the reporter (especially on social media), thus decreasing the level of anonymity.</p> <p>The app has a lot of potential for automation. As it stands now, investigation and any retribution (as it were) must be carried out entirely through humans, hampering its effectiveness. Since it requires human effort (which we humans hate doing), this might further detract from its effectiveness.</p> <p>A key failing in the UI is that the method for uploading evidence is complicated. It requires the user to upload an image, which (in the case of cyberbullying) requires taking a screenshot. This is quite complicated on a phone, and detracts from the usability.</p>	<p>Aesthetics: 7/10</p> <p>Ergonomics/UX: 4/10</p> <p>Functionality: 6/10</p> <p>Safety: 7/10</p>
Stop!t	<p>One of the schools which adopted the pilot found that it acted as a deterrent- the school received 75% fewer reports of bullying (Lapowski, 2015). While correlation famously does not imply causation, it does look promising.</p> <p>Evidence or information can be added in the form of videos, text, photos, etc. The use of text is particularly useful in the context of cyberbullying, since most cyberbullying takes the form of vitriolic insults. Also, copying text is more convenient than having to take and edit a screenshot on a phone.</p> <p>The app has clearly delineated functions for students and teachers, such as allowing teachers access to aggregate data and analysis. This not only makes it more secure, preventing students from accessing potentially sensitive data, but also makes it more usable for teachers and admins who have opportunities to be proactive.</p> <p>It also has functions for 2-way messaging which builds upon the reporting function, such as allowing students to follow up on complaints.</p> <p>The app also has an excellently calming colour scheme, being mostly</p>	<p>Requires publicity to be used as a deterrent; otherwise no one knows about it and it is useless, both for bullies and potential users. However, announcing the app can make it so widely known that it becomes difficult to use. For instance, it might lead to a deluge of spamming; false accusation might instantly become a new form of bullying among students.</p> <p>Evidence must be added manually, in the form of a screenshot for cyberbullying. This is particularly hard on a phone/tablet; ideally it should be able to extract the offending post and embed it in the report. The way the evidence is structured also allows for falsification.</p> <p>The app requires an access code to use, which is likely lead to a lot of administrative complications. This not only decreases its potential but also diminishes the User Experience; especially for users who do not have the code, and will need to jump through hoops for it.</p> <p>The UI on the backend is sprawling; there are too many options to easily navigate. This could be remedied using</p>	<p>Aesthetics:8/10</p> <p>Ergonomics/UX: 5/10</p> <p>Functionality: 7.5/10</p> <p>Safety: 5/10 (It still has options which can be used to damage the reputations of other students).</p>

	<p>in blue. The client-side UI is also well made, being very simply organised into reporting and anonymous messaging.</p>	<p>drop-downs or sub-menus, or simply streamlining it and eliminating superfluous features.</p>	
Rethink	<p>The app is (in principle at least) extremely simple both from a systems engineering point of view and from the user's end. This is useful because it means that the app is easy to sell and utilise for students, and because it comes with less potential for things to go disastrously wrong.</p> <p>The app prides itself on being non-intrusive. This minimalism also makes it more attractive to users, and thus more potentially effective. It only provides a little bit of friction, but is nonetheless extremely effective, just by providing a tiny bit of friction.</p> <p>The key feature of the app is that it prevents cyberbullying before it can happen, on the bullying end rather than on the victim's. This is valuable because it can (in a sense) reform bullies by speaking directly to them. Since the software is completely self-contained it removes potential for human error or malice, such as spamming; there is no human system which can be overwhelmed, fooled, or spammed.</p> <p>The app is always on</p> <p>The UI of the app is generally minimalist, since it replaces the keyboard. It does not need to be activated to be effective.</p>	<p>Not strictly a disadvantage, but it is targeted more at prospective bullies than at victims. Thus, while it is worth analysing the amount I can learn is somewhat decreased.</p> <p>Since it ultimately just provides a choice, the app might not affect determined or chronic bullies, who would likely simply ignore it.</p> <p>The app also does not have a discernible effect on the victims, who are ultimately the people most negatively affected by cyberbullying. It also does not provide any outlet for external analysis or moderation such as from teachers or automated systems.</p> <p>The filter says that it scans for offensive terms. Thus, it is likely to fall victim to the Scunthorpe problem, of giving false positives and being easily fooled by simple human workarounds, thus impairing its utility. For instance, people are likely to simply switch to using slang/ unofficial terms which cannot be recognised by the computer, but can be understood by people.</p> <p>A major failing is almost by design- it is designed to work against the user rather than with them. Thus, it is vulnerable to the age-old workaround of deleting it or turning it off.</p> <p>Another key problem is that the prompts given get irritating after some time.</p> <p>It is also inconvenient because it erases everything, not just the offensive word- this de-incentivizes users from accepting the correction, even if they acknowledge it.</p>	<p>Aesthetics: 6/10 Ergonomics/UX: 4/10 Functionality: 5/10 Safety: 8/10</p>
Tootoot	<p>The app is more diverse than the other ones on this list. Firstly, it is designed for a wider audience; it has features designed for parents and guardians as well as for teachers and students. Since it facilitates communication between the various groups as well, the app helps them coordinate their efforts (as it were). This also increases its potential for gaining useful data, and thus greatly increases its long-term effectiveness.</p> <p>The app is also very diverse in its functionality; it is designed for students to air any concerns they have, whether academic or social. Since it is not tied down to bullying, it might also</p> <p>Analysis and reports</p>	<p>Not focused on bullying, which likely decreases its effectiveness in that sphere, since it probably lacks the specific support systems for that kind of behaviour.</p> <p>The service acts on an extremely large scale, meaning that it might introduce the various trappings of bureaucracy and inefficiency. Since it attempts to cover multiple schools as well as a large amount of problems, it is likely not tailored to any of them, which weakens its potential. This lack of specificity is a weakness, since it means that the app sacrifices depth for breadth.</p> <p>The app waits only until something is typed, not necessarily posted.</p> <p>It seems only to track swear words.</p> <p>The app is keyed solely to typing, and</p>	<p>Aesthetics: 6/10 Ergonomics/UX: 7/10 Functionality: 8/10 Safety: 7/10</p>

	Ensures confidentiality of information between students, teachers, parents. This helps with safety scores.	does not discriminate based on where it is typed. Even when making notes, etc. it still registers.	
--	--	--	--

Primary research

Questions for Ms Rekha:

Questions	Answer summary
What sort of data might be useful?	The type of cyberbullying The account of the offender The account of the victim (if possible) The platform and if possible device Basic demographics such as age of the offender
What do you think could be done with the reports?	Likely send an automated email to the offender, and open a dialogue suggesting that they make amends, suggest gently that they apologise to the victim, etc.
How can we make sure that the service isn't abused?	Make evidence compulsory, and scan for offensive terms in submitted posts.
What are the main forms of cyberbullying to focus on?	Harassment Rumour mongering Exclusion In general, those which impact both self-esteem and reputation.
Is there a way to limit this to RWA students? Should we do so?	The best way is to only 'advertise' it to RWA students, and possibly only make the software available from the RWA website or something.
Generally, what improvements do you think can be added?	Have an attractive background and design on the server side as well, including attractive designs and inspirational quotes, rather than just a database.

Questions for Shaffaf (technical person and acquaintance of my father)

How can I create the backend for a website with minimal programming?	If you just need a database, you can use google firebase. It allows for storing, writing, and reading data, and works in real time. Just visit the site and start reading the documentation.
--	--

Audience Survey (heavily summarised and generalised- full response data and proof in appendix A, including charts and lists)

What sites do you experience/see cyberbullying on?	Youtube, Facebook, Instagram, sometimes twitter
Would you like a simple UI, or a more complicated one with more options?	Somewhere in between, slightly more minimalist
If you reported an incident of bullying, what facts would be relevant/important?	Platform, people involved, URL, date, record of things said, context/history
Should report forms be presented as web, email, or some other?	Web form, probably
Should the layout be like a traditional web form, or more conversational?	Ideally more conversational
What colour scheme would you suggest for the app? a) High or low contrast b) Warm or cool colours? c) Minimalist or colourful?	a) Low contrast b) Cooler tones c) Minimalist, but not too much so

Primary technical research (experimentation)

Question	Answer	Methodology
How can a program identify the text of a facebook post for extraction	Enclosed within a <div> with class="_5pbx userContent _3576"	Selected a post from the appropriate site, and used the chrome 'inspect'

	The user name is contained within an <a> tag with class _2wmb.	element' function to determine the class type. Double checked with at least 3 other posts, from varying users, to make sure that the system was generally reliable.
How can a program identify the text of a twitter tweet?	Enclosed within a <p> tag, with class TweetTextSize TweetTextSize--jumbo js-tweet-text tweet-text. The username is the first instance of a tag with class ="username u-dir u-texfTruncate"	
How to extract an Instagram image post?	Enclosed within <div class="eLAPa kPFhm">. Alternatively, they offer options for sharing links, which can be done on the user-end.	

Secondary research plan (answer summaries and referencing can be found in appendix A):

Priorities: 1-5 (5 is highest)

Question	Priority	Source
What is JavaScript, and how does it work?	5	https://www.w3schools.com/js/ https://www.w3schools.com/jquery/jquery_syntax.asp
How do I go about creating a chrome extension?	5	https://developer.chrome.com/extensions/getstarted https://developer.chrome.com/extensions/overview
Can I extract text from a tweet using JavaScript?	3	https://github.com/twitter/twitter-text/tree/master/js https://api.jquery.com/first-selector/
How can I fetch a twitter post for sending via email/website?	3	https://cypressnorth.com/programming/how-to-easily-display-a-twitter-feed-using-javascript/
Is there a way to send automated emails using JavaScript?	3	https://www.w3schools.com/nodejs/nodejs_email.asp https://knowledge.hubspot.com/articles/kcs_article/cos-pages-editor/how-do-i-create-a-mailto-link
How can I search for and highlight specific text in a webpage?	4	https://www.encodedna.com/jquery/make-jquery-contains-selector-case-insensitive.htm https://api.jquery.com/
What sites do people frequently experience cyberbullying on (Stats)	3	https://www.ditchthelabel.org/wp-content/uploads/2017/07/The-Annual-Bullying-Survey-2017-1.pdf
What are some signs of cyberbullying in a text that a computer can search for?	1	https://socialwork.tulane.edu/blog/cyberbullying-awareness-guide
How can I search for cussing/ offensive words in a document text?	3	https://github.com/MauriceButler/badwords https://www.w3schools.com/jsref/jsref_regexp_test.asp https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions
Are there any patterns in offensive messages which can be detected using computer analysis?	4	http://webpages.ursinus.edu/akontostathis/KontostathisWEBSCI2013.pdf
Syntax for checking for a site/URL	5	https://gist.github.com/prasanthj/a5b71c47ee0dec3bbb72 https://developer.chrome.com/extensions/tabs

How do I get the URL of the current tab	4	https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API/tabs/query https://developer.chrome.com/extensions/tabs
How can I run a script in a tab (without opening my own HTML)	5	https://developer.chrome.com/extensions/tabs
Objective criteria and levels of cyberbullying, for classification and possibly retribution on the server-end.	4	http://www.northlamar.net/upload/common/18-19%20Required%20Docs/Bullying%20Checklist.pdf
How can I prevent users from spamming the site with random tweets?	3	http://stopitsolutions.com/blog/false-reporting-busting-myths/
Different libraries and permissions for chrome extensions	5	https://developer.chrome.com/extensions/api_index
What sort of display would be apt. for my UI?	4	https://www.smashingmagazine.com/2017/06/designing-efficient-web-forms/
What would be a good, calming colour scheme for my app?	3	https://uxplanet.org/color-theory-brief-guide-for-designers-76e11c57eaa
Cyberbullying background: What might drive someone not to report an incident?	4	https://www.verywellfamily.com/reasons-why-victims-of-bullying-do-not-tell-460784
How can users report anonymously?	5	https://www.w3schools.com/html/html_forms.asp
What does API mean?	2	https://www.programmableweb.com/api-university/what-are-apis-and-how-do-they-work
What kinds of students generally experience cyberbullying/abuse, and in what proportions (i.e. quantitative answers).	2	http://facecrooks.com/Cyberbullying-internal/Cyberbullying-Statistics.html https://www.teensafe.com/blog/cyber-bullying-facts-and-statistics/
What are the various forms of cyberbullying?	3	https://www.avvo.com/legal-guides/ugc/10-types-of-cyberbullying
How can I design a good icon for my app?	4	https://www.smashingmagazine.com/2009/08/vital-tips-for-effective-logo-design/
How can I write data to Google Firebase?	5	https://www.tutorialspoint.com/firebase/firebase_write_list_data.htm
How can I get the contents of an element using jquery?	3	https://www.w3schools.com/jquery/trysel.asp
Is it possible to store variables across pages in chrome?	4	https://developer.chrome.com/extensions/storage
How can I read and sort data from Google Firebase?	5	https://www.tutorialspoint.com/firebase/firebase_read_data.htm
How do HTML forms work?	3	https://www.w3schools.com/html/html_forms.asp

Ideas	Requirements
Immediately be accessed from a page with an offending post	Event.listener' code, so that the form can launch whenever a button is clicked.
Automatically embed text from posts	JQuery library, specifically functions for searching and parsing. If the user does it correctly, this can be done using getelement functions, and by searching for a <p> tag within the script (text posts on a webpage are almost always stored within <p> tags).
Have users submit data in a web form	An HTML webpage with a <form> tag, and a 'POST' system.

Search submitted reports for swear words and signs of bullying	Using ctrl+f while accessing firebase in a browser and a basic text search function (JQuery again, presumably) equipped with regular expressions.
Access to user URL and ability to execute scripts/ pop-ups in Chrome	Access to Chrome developer API's, which are fortunately easily accessed.
Storing data	Access to a Google firebase database, and a way of writing/reading data to and from it.
Data analytics on the server side	Google offers a service called Firebase which allows users to create a JSON database for their app which can be read from and written to. Firebase also has options for queries, or the JSON data can simply be analysed via a separate program.
The users must be able to submit reports anonymously	Submit data via forms rather than via mail, and don't ask for any personal information (or at least not make it a compulsory field).
Preventing spam	A compulsory field for evidence, such as link/embedding for offending posts. Data validation using JS, such as 'type of' command which checks the datatype of a variable.

Design Brief

The product I have ultimately chosen to create is an extension for google chrome. The extension will allow victims or witnesses of cyberbullying to report incidents directly to the ABC. The key feature of the app is that it will require some sort of evidence, usually in the form of a link to the offending post. This feature is very much central to my idea, since it eases the load on the administrator's side, and prevents users from spamming or making false complaints.

Since a Chrome extension is essentially a website, I can create it primarily using HTML. However, since it is fundamentally interactive, extensive use (and knowledge) of some sort of web programming language is necessary. I have chosen JS over other web development languages such as PHP because it has the most support in HTML (being the web standard). Additionally, JavaScript has far more extensions and add-ons available (such as JQuery) and is thus easier to develop in, since plug-ins such as nodejs make it possible to write an entire web app (including server-end) entirely in JS. For the backend of my website I intend to log reports in Google Firebase, where each report will have a unique key and will be able to be queried.

The target audience of my product is intended to be victims of cyberbullying, specifically in Raffles World Academy. It is directed at all students, but I intend it to be more prevalent among secondary students, since cyberbullying is far more common among teenagers.

The overall intention of the project is twofold:

- a) To provide victims of cyberbullying with an alternative to staying quiet and encourage them to speak out: Anonymously reporting to a group of students is far easier than verbally telling a teacher; this extension should make reporting cyberbullying (especially as a victim) less stressful and rare. Creating this extension also makes it clear that cyberbullying is a serious issue worth reporting.
- b) to help the ABC become informed about cyberbullying in the school, and by extension be better equipped to prevent it: The main difficulty with dealing with cyberbullying is that it cannot be handled or reported conventionally. Thus, it is almost impossible to know much about it; hence why I am using a database to keep track of and analyse reports for the ABC.

Based on my research, I know that most public cyberbullying happens on massive social networks. Smaller networks and forums typically have dedicated moderators, who do a fair job of preventing abuse. Facebook and Instagram are the primary offenders (more than a third of students report some cyberbullying on Facebook). Per my primary research Facebook use has declined significantly, meaning that twitter and Instagram will be my primary foci. Thus, along with general reporting, my extension needs to have specific functions to make it more usable on this 'unholy trinity'. I intend to automate the retrieval of offending posts and post content for these sites, to save on effort and encourage users to report. I know that this can be done, using some complex functions which I intend to borrow from GitHub, with due credit (since they are open source). This has the added advantage of making it exceptionally easy for the ABC to view and authenticate reports, since all information is right in front of them in text form- which can also be analysed by a server-side program which can scan for offensive words. Depending on the presence and number of swear words in the media text, the reports will be prioritised when viewed by a server/admin.

The app will essentially be a button for users to press whenever they see cyberbullying. Once the button is pressed, it should launch a basic HTML webpage with a form for them to fill in, which includes things like perpetrator (account name), description of incident, URL (the most important), etc. It will also include optional fields for personal data such as name (of the reporter), which will have an option for anonymous or blank. The form will have in-built basic validation techniques, such as double-checking that the URL is valid and the data types match.

Most of the fields within the form will be automatically filled, using data-extraction techniques and HTML substitution. The URL field will be automatically filled, based on the URL from which the extension was activated (which will be written to and retrieved from chrome storage). The 'offending media' field, which includes the text or image deemed offensive, will also be automatically

filled. JQuery has a variety of ‘find’ and ‘contains:’ commands which can be used to obtain the text of a post and then fill it into the appropriate field.

The logo/extension button of the app should be extremely simple; the most probable design will be a dark blue hand, inspired by the ‘Stop bullying’ hand logo. To be appropriate to the small size, it will be heavily stylised, possibly looking like the hand from the Gravity Falls Journal. The form itself will have a dark-blue and white colour scheme, which should ideally have a calming effect on users.

Criterion B

Design specifications

Technical/Misc.:

- Retrieving the evidence in the form of URLs and/or text should be automated for major social media sites (Twitter, Facebook, Instagram).
- Report data must be stored in SQL or some web database, so that it can be logged or analyzed
- The product should be able to run perfectly as a chrome extension
- It should use efficient coding and minimal media to load quickly.
- It should be equipped to deal with all the various types of cyberbullying: harassment, rumor-mongering, flaming, etc.

UX (aesthetics and ergonomics) (Will be evaluated by client survey)

- The app should be easy enough to use that only a few sentences of instruction are necessary; ideally, it should be completely intuitive.
- The form fields should be unambiguously labelled
- Anything relating to evidence of the incident should be automatically filled
- Evidence of the incident reported will be mandatory, in the form of text/image and a URL.
- There must be an option for anonymous reports
- The form should have a minimalist design, with only a few colors/ styles, and with unity of effect.
- The form should have a calming blue-white color scheme to put users at ease

Logo/Icon:

- The logo should be small enough for the extension bar
- It must have a bold color to stand out.
- It should give a good idea as to the purpose of the app

Server-side

- The report data should be easily accessible by the admin users, and not require complex retrieval.
- A way of notifying admins when complaints are made

The overall aim of my project is to provide students with a way to report cyberbullying. This means that the users must enter data in some kind of form, since that is the easiest both for them to enter and for the server to process.

Design 1: Minimal formatting, and sends form via email

Cyber-bullying report #123

serveraddress@rwadubai.com

Cyber-bullying report #123

Details

Name (Optional):

Type of Bullying (Optional):

Incident Description: (2-3 sentences, to be filled by the user)

Evidence

URL: example.com/posts/offensive-post

Offending user: @XYZ

Platform: Twitter

Message: 'I cannot but conclude the bulk of your natives to be the most pernicious race of little odious vermin that nature ever suffered to crawl upon the surface of the earth.'

Annotations:

- An arrow points from the text "Automatically set, to whatever the server address is chosen to be." to the recipient field "serveraddress@rwadubai.com".
- An arrow points from the text "The number is also set automatically" to the subject line "Cyber-bullying report #123".
- An arrow points from the text "Anonymous reporting is an option" to the "Name (Optional)" field.
- An arrow points from the text "Subheadings, which group the fields" to the "Details" section header.
- An arrow points from the text "This information is all filled in automatically, using JavaScript. It is included in the form for server-side convenience." to the "Evidence" section header.
- An arrow points from the text "Sans-Serif font, size 12 (as is standard for email). The field names are in bold, and the subheadings are bold, italicised, and underlined." to the "Message" text area.

(Created in Gmail, annotated in Preview)

Design 2: Web Form: Once again, evidence will be auto-filled using JavaScript.

Cyberbullying Report #X

Details

Name (Optional):

Type of Bullying:

- Flaming (Vulgar or insulting language)
- Denigrating (Gossiping or spreading rumors)
- Cyber harassment (Offensive and threatening messages)
- Impersonation (Pretending to be someone else)
- ‘Outing’ (Revealing private information)
- Unknown
- Other: _____

Incident Description:

Evidence

URL:

example.com/posts/offensive-post

- Correct

Offending User:

@Jonathan_Swift

- Correct

Platform:

Twitter

Message:

'I cannot but conclude the bulk of your natives to be the most pernicious race of little odious vermin that nature ever suffered to crawl upon the surface of the earth.'

- Correct

Report

The title is in size 26 and bold for clarity. All fields/subheadings are in Georgia font, while answers and options are in Garamond.

Multiple choice question, for user ease (using radio buttons).

Options are in Garamond size 12 (thinner and smaller)

Subheadings:
Size 16 font,
underlined

Georgia size 13

Tick boxes for the user to confirm these inputs

Garamond size 13
(comparatively thinner)

Report/submit button, bordered and elevated with 3d-effects. Text is bold, size 14.

Design 3: No-effort (for the user)- the only thing they do is check the data and submit it.

Cyberbullying Report

URL: _____

Offending Account: _____

Platform: _____

Message: _____

Submit

Size 16, Italicised

The text is all in font Georgia (a traditional web font).

The information in the fields will be filled automatically on the blue lines using JS

On submitting, the webpage will display a confirmation prompt in the form of a dialog box (window.alert), asking 'Is the information correct?'

On confirmation, the form is sent to the server.

Design 4: Conversational tone (Questions and answers are in speech bubbles):

What's your name? (Feel free to remain anonymous)

What type of bullying was this?

Could you describe the incident briefly?

Uniform font of sans-serif family, size 12.

Questions and answers in slightly different shades (dark and light blue).

Phrased in conversational format, for user comfort.

Vulgar/rude language
 Gossiping
 Harassment
 Impersonation
 Revealing private info
 Other: _____

Multiple choice, using tickboxes.

Was this the message? (Correct any errors, please)

The other fields (URL, etc.) are automatically filled, and are invisible elements of the form

'I cannot but conclude the bulk of your natives to be the most pernicious race of little odious vermin that nature ever suffered to crawl upon the surface of the earth.'

What was the poster's account?

@Jonathan_Swift

Automatically filled.
This is just verification/redundancy.

Auto-submits once the last question is answered and entered.

Evaluation against specifications:

Specification	Design 1	Design 2	Design 3	Design 4
The app should be easy enough to use that only a few sentences of instruction are necessary; ideally, it should be completely intuitive.	✓	✓	✓	✓
The form fields should be unambiguously labelled	✓	✓	✓	✓
Anything relating to evidence of the incident should be automatically filled	✓	✓	✓	✓
Evidence of the incident reported will be mandatory, in the form of text/image and a URL.	✓	✓	✓	✓
There must be an option for anonymous reports	✗	✓	✓	✓
The form should have a minimalist design, with only a few colors/ styles.	✓	✓	✓	✓
The form should have a calming blue-white color scheme to put users at ease	✗	✓	✓	✓

Evaluation: Pros and Cons

	Design 1	Design 2	Design 3	Design 4
Me	<p>Very simple to implement and manage; it only requires an email ID and use of the HTML mailto: link.</p> <p>Easy for users, since they know how emails work.</p> <p>Does not require any complex server-side scripting, since data is automatically logged as an email ID</p>	<p>The color scheme is attractive and looks quite professional.</p> <p>It strikes a good balance between detail and efficiency, by having some auto-filled but not all.</p> <p>The style is unified throughout the form.</p> <p>This design fulfills all the outlined design specifications.</p> <p>The serial number in the form title is useless and also complicates my coding;</p>	<p>Very easy for users, removing friction for submission- they have to do virtually nothing.</p> <p>The UI would be very easy to design.</p> <p>The UI is also very clean and minimalist, and the various elements are clearly visible.</p> <p>Lacks any true user input-meaning factors like context might go ignored, making it difficult to be accurate.</p>	<p>Far more natural for users, and should generally be a better experience.</p> <p>Much less clichéd than conventional forms, and thus a less boring UX.</p> <p>The use of invisible fields makes it much more streamlined.</p> <p>There is no clearly defined submit button, which might confuse users.</p> <p>Users might enter data in conversational style (such as using sentences) which</p>

	<p>Arguably too minimalist; there are no colors or styling and it looks extremely sterile and dull.</p> <p>Fails many of the design specs.</p> <p>No method of data validation possible.</p> <p>Very impersonal, no UI to speak of.</p> <p>Likely inconsistent depending on the user's mail client and account—the way the data is transmitted might vary, making it difficult to process.</p> <p>Doing data logging and analysis (as I intend to) would be extremely difficult and require access to all sorts of mail APIs.</p>	<p>keeping track of data can be done based on timestamps and keys in the database.</p> <p>Some of the details required might be superfluous, such as the URL (it could probably be hidden and auto-submitted).</p> <p>The high amount of user input might discourage users from reporting.</p> <p>The form design might be a little old hat and boring for users.</p>	<p>Since the type of bullying is not specified, this impairs the ability to sort and analyze data.</p> <p>Users might ignore double-checking the auto-filled data, which would likely lead to errors (especially on Instagram, since it organizes text differently to most other sites).</p> <p>Looks unsettlingly like a Microsoft form, which would probably lead to some confusion.</p> <p>Quite frankly it would be far more convenient to just have the info submitted automatically (since the user does not provide any data).</p>	<p>makes record keeping impossible.</p> <p>More appropriate to a phone than a laptop or desktop.</p> <p>Difficult to accomplish in HTML and/or CSS, especially not for creating a form.</p>
Client	This could work, but it looks a little bit dull. Also it might be complicated to deal with.	This design is my favorite I think. It's very detailed and looks quite professional and formal. It strikes a good balance between being detailed and being user-friendly.	Not a very good design. A bit lacking in details, and generally quite bland.	The design is nice and user-friendly, but it seems a bit too informal for the purpose. The idea of using very accessible language is a good one though.
Final decision	<p>My final chosen design is some variation on design 2. I made this choice primarily through the process of elimination; the first design failed the specifications, and the fourth one was effectively impossible (at least with the basic tools I have at hand). The third design was certainly practical, and quite elegant from a user point of view; but there was not much data to collect which impaired its use. The second design solves these problems by being fairly extensive. Furthermore, it is the one favored by my client for its professional look.</p> <p>However, this design introduces the problem of being annoyingly long, which must be remedied in some way. This means that the form needs to be trimmed down a bit, which can be partially done via that great panacea of automation.</p> <p>The main alterations I intend to make are various forms of streamlining. The URL and platform fields will be made invisible (using CSS display=none). This will also eliminate the need for subheadings and thus stop the form seeming so bulky. These fields will still have to be in the form in some way, in order to be submitted to the database properly. I might also rephrase many of the fields in a question format like in design 4, to make it a more natural and accessible experience.</p>			

Final Design:

Cyberbullying Report

What is your name? (Feel free to remain anonymous)

Title is in size 26, bold, Georgia.

What type of bullying was this?

- Vulgar/rude language
- Gossiping
- Harassment
- Impersonation
- Revealing private info
- Other: _____

Background colour is #006699 in hexadecimal.
Text is white (#ffffff)

Could you describe the incident briefly?

Question text is in Georgia, size 13, sentence case.

What was the poster's account?

@Jonathan_Swift

Was this the message? (correct any errors please)

'I cannot but conclude the bulk of your natives to be the most pernicious race of little odious vermin that nature ever suffered to crawl upon the surface of the earth.'

Options will be in Garamond size 12, and will be marked with radio buttons (multiple choice).

Answer text will be in Garamond size 13 (thinner than the questions but still clearly readable). It will use a textarea to emulate large input.

User has options to correct text, in case of errors

Report

Report/submit button, bordered and elevated with 3d-effects. Text is bold, size 14.

To confirm that the form was submitted, it will display a dialog box which says 'Form submitted. Thank you for the report' or some such once the report button is clicked.

Logo/Icon designs:

Idea 1: Journal hand in blue (six-fingered to stand out):



Idea 2: Red 'stop' logo (Circle and cross), with a B:



Idea 3: A speech bubble, to symbolize speaking out



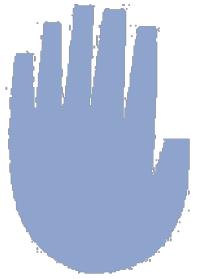
Evaluation against specifications:

	Design 1	Design 2	Design 3
The logo should be small enough for the extension bar	✓	✓	✓
It must have a bold color to stand out.	[Debatable]	✓	✓
It should give a good idea as to the purpose of the app	✓	✓	✓

	Design 1	Design 2	Design 3
My evaluation	<p>It has precedent; the 'stop bullying' hand is a well-known and common logo.</p> <p>The 6-fingered hand is, I feel, a nice gimmick which adds some originality to the logo.</p> <p>The logo coloring is somewhat dark, and may not stand out against the chrome background (depending on</p>	<p>The color is very bright and stands out clearly.</p> <p>The logo makes the purpose of the app completely clear, with the minimum of fuss.</p> <p>It is not particularly detailed, and thus should work well as an extension icon.</p>	<p>This provides a very clear idea of what the extension is meant to do, as a reporting software.</p> <p>The logo should be easy to create/render, since it is basically just a single shape.</p> <p>The design is also quite simply and heavily stylized.</p>

	the color theme on the user-end)	The logo might be construed as generic, since the 'ban' symbol is quite common. The text might be obscured or misunderstood, since it is only a single letter and since the image is generally extremely tiny Rendering the combined image might be a problem since my graphic design skills are lacking to the point of non-existence.	Despite being simple, it is unlikely to be confused for some other extension. It lacks originality and creativity; the logo is somewhat cliché.
Client evaluation	This is probably my favorite design. It makes the purpose very clear, it looks simple, and it also has quite an attractive color which goes well with the form design.	It would be difficult to see in such a small size, but otherwise it is good, especially the bright colours.	This one is quite good, but I think the others are more appropriate to the app.
Final decision	My final decision is to go with the blue hand, mainly because my client suggested it. This style of logo is also well-established as an anti-bullying symbol (stopIt, which I have analyzed, uses a version of it as a logo).		

Final icon design



Server-side UX:

This lies outside of the scope of my project/problem, since this one is more focused on users than on the back end. I intend to handle this aspect with the basic firebase display for now. The firebase architecture is quite easily accessible, since it can be searched (via co-opting the Ctrl+F function in a browser) and the records can be easily compressed or expanded. The firebase records are also stored in chronological order (using auto-generated keys) and can be timestamped using the JS date() function, which meets most of the design specifications.

Criterion C

The GitHub repository from where my project can be downloaded (both as a folder and as a chrome extension file):
<https://github.com/Rohan-Goyal/cyberbullying-reporter>

Hardware resources required: MacBook pro

Software resources required: Visual Studio, Google Chrome, Google Firebase, Chrome and Firebase documentation

Tools required:

Visual Studio:

- JSON creator and editor (for the manifest file)
- HTML creator and editor (for the form webpage)
- CSS creator and editor (for the form stylesheet)
- JavaScript creator and editor (for creating JS files)

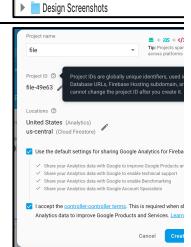
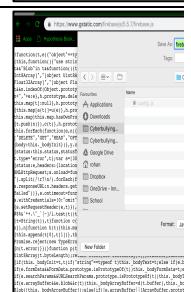
Google Chrome: developer mode

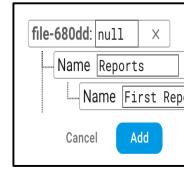
Finder:

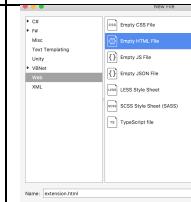
- Find folder
- Create new folder

Note: The time taken for the various 'testing steps' should essentially be 1-2 minutes. I have allowed some more considering the possibility that I need to remedy my errors (which I shouldn't, especially since I have a detailed and rigorous plan).

Note: In case of any confusion as to how something should look like, c.f. the larger screenshots for every step (which can be found in appendix C).

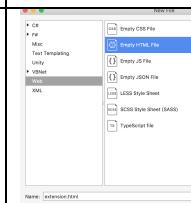
Step	Sub steps/ actual code commands	Resources	Time (mins)	
Create the extension folder	In Finder, navigate to OneDrive- innoventures education, and select 'new folder'. Name it 'Cyberbullying Reporter'.	Finder	1	
Sign up for Google firebase and create a database	<ul style="list-style-type: none">• Go to the google firebase website• Sign up with google account• Name the project 'file'.• Read and accept the terms and conditions• Click 'create project'	Firebase Chrome	3	
Get the firebase initialization code	<ul style="list-style-type: none">• Click the 'web app' option on the get started screen.• Copy the configuration code into a new JavaScript file (in Visual Studio) and save it to cyberbullying reporter as config.js. (since Chrome extensions do not allow inline scripts).• Go to the script source (https://www.gstatic.com/firebasejs/5.5.7/firebase.js) and using ⌘+S save the file to 'Cyberbullying Reporter' as firebase.js	Firebase Chrome	5	

Create a realtime database in Firebase. Add a child node 'reports'	<ul style="list-style-type: none"> Go to the database option in the Firebase app Scroll down and select the real-time database option. For convenience sake, set it to testing mode. Under the project ID, select the + and add a child 'Reports'. Below reports, add another child with name 'first report' and value 'hello world'. 	Firebase Chrome	4	
Download and save jQuery	<ul style="list-style-type: none"> Go to https://ajax.googleapis.com/ajax/libs/jquery/1.8.1/jquery.min.js Save the file as jquery.min.js using ⌘+S (again, save it to Cyberbullying Reporter folder). 	Finder Chrome AJAX	1	
Get the logo image	<ul style="list-style-type: none"> Right-click the final logo in crit B of this document. Select save as picture Save it as speech_bubble.png to the cyberbullying reporter file 		1	
Create the extension manifest (JSON file)	[General notes, not steps] be sure to include commas after every value (except the last) and that each 'sub-field' should be enclosed in its own parentheses	Visual Studio	16 (in all)	
Create the JSON file	Open visual studio and press ⌘+n. Go to the 'web' section and select 'new JSON file'. Type 'manifest' into the name bar.		2	
Define the name, version, description	<ul style="list-style-type: none"> Place the cursor between the {} brackets. Type the code "name": "Cyberbullying Reporter", "version": "1.0", "description": "Report cyberbullying directly to the ABC", 		2	
Define the permissions (for all URLs and for some Chrome APIs)	"permissions": ["webNavigation", "tabs", "windows", "activeTab", "storage", "https:///*/*"],		2	
Define the background scripts, which are always running/listening for events.	<ul style="list-style-type: none"> "background": { "scripts": ["background.js", "firebase.js", "firebaseCDN.js"], "persistent": false }, 		1	
List the browser action icons (clickable things)	<ul style="list-style-type: none"> "browser_action": { "default_icon": { "16": "speech_bubble.png", "32": "speech_bubble.png", "48": "speech_bubble.png", "128": "speech_bubble.png" } }, 		3	

Specify the icon sources for different sizes (all the same)	<ul style="list-style-type: none"> • "icons": { • "16": "speech_bubble.png", • "32": "speech_bubble.png", • "48": "speech_bubble.png", • "128": "speech_bubble.png" • }, 		3	<pre>"icons": { "16": "speech_bubble.png", "32": "speech_bubble.png", "48": "speech_bubble.png", "128": "speech_bubble.png" },</pre>
Specify the list of web accessible resources (which can be called and run by the extension scripts)	<ul style="list-style-type: none"> • "web_accessible_resources": [• "extension.html", • "firebase.js", • "config.js", • "firebaseCDN.js", • "function.js", • "inject.js", • "style.css", •], 		4	<pre>"web_accessible_resources": ["extension.html", "firebase.js", "config.js", "firebaseCDN.js", "function.js", "inject.js", "style.css",]</pre>
Identify manifest version	"manifest_version":1		1	<pre>"manifest_version":</pre>
Create the HTML page		Visual Studio Chrome	30 (In all)	
Create the HTML form, and load the various scripts that are necessary.	<ul style="list-style-type: none"> • Open visual studio and press ⌘+n. • Go to web, and select empty HTML file'. • Name it extension.html. Within the head, insert the tags: • <script src="jquery.min.js"></script> • <script src="firebase.js"></script> • <script src="config.js"> </script> to load the firebase and jquery dependencies. • <link rel="stylesheet" href="style.css"> • Fill in the title tag as 'Cyberbullying Report'. • Insert a form tag with method attribute='post' • (all CSS will be defined later) • The rest of the head tags are autocreated (as well as the body tag) upon creating an HTML file in VS. 		5	
Create the name field and question	<ul style="list-style-type: none"> • Create a name field as follows: • <h3>What is your name? </h3>
 • <input type="text" name="name" id="name" />
 • <h2> Feel free to remain anonymous</h2> (h2 will make it a subtitle field, to be defined in the CSS)
 		1.5	<pre><h3>Name:</h3> <input type="text" name="name" id="name"> <h2>Feel free to remain anonymous</h2></pre>
Create the description field and question	<ul style="list-style-type: none"> • Could you describe the incident?
 • <textarea name="description" id="description" />
 		1.5	<pre><h3>Could you describe t
 <textarea id="descriptio</pre>
Create the account field and question	<ul style="list-style-type: none"> • What was the poster's account?
 		1.5	<pre><h3>What was the poster
 <input type="text" name="</pre>

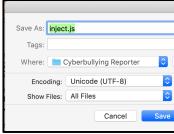
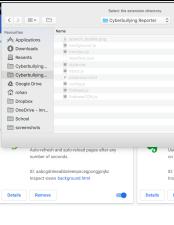
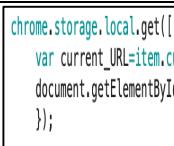
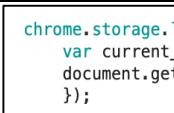
	<ul style="list-style-type: none"> • <input type="text" name="account" id="account" />
 		
Create the message field and question	<ul style="list-style-type: none"> • What was the message?
 • <textarea name="message" id="message" />
 • <input type='checkbox' id='correct_message' name='correct_account'> •
 	1.5	<pre><h3>What was the message?</h3> <textarea id="message" rows="5">

</textarea> <h2>(Correct any errors)</h2></pre>
Insert the multiple-choice question	<p>Just after the </h2>
 tag, insert the following code:</p> <ul style="list-style-type: none"> • <select id='type'> • Insert the options here: • <option value='vulgar/rude language'></option> • <option value="Vulgar/rude language">Vulgar/rude language</option> • <option value="Gossiping">Gossiping</option> • <option value="Harassment"> Harassment </option> • <option value="Impersonation"> Impersonation</option> • <option value="Revealing private info">Revealing private info</option> • <option value='Other'>Other/Don't know</option> • </select>
 	5	<pre><select id="type"> <option value=""></option> <option value="Vulgar/rude language">Vulgar/rude language</option> <option value="Gossiping">Gossiping</option> <option value="Harassment"> Harassment </option> <option value="Impersonation"> Impersonation</option> <option value="Revealing private info">Revealing private info</option> <option value='Other'>Other/Don't know</option> </select>
</pre>
Create the (invisible) URL and Platform fields.	<ul style="list-style-type: none"> • <p id='URL'></p> • <p id='Platform'></p> • </form>, closing the form • (Used a <p> tag for convenience, since it is invisible). <p>The tags will be filled using JS variables later, once the code is created and run.</p>	2	<pre><p id="URL"></p> <p id="platform"></p> </form></pre>
Create submit button	<ul style="list-style-type: none"> • Insert a <button> tag outside of the form tag, with id='submit'. • Within the button tag, place the text 'report'. • Outside the button insert the tag <script src='function.js'></script> • Outside the script tag, close the body and html tags. • The HTML page is now complete. 	2	<pre><button id="submit">Report</button> <script src="function.js"></script></pre>
Testing	<ul style="list-style-type: none"> • Save the webpage to the folder 'Cyberbullying Reporter' with the extension .html. • Navigate to the file in finder and double click. • The app should open in Chrome. • If it does not look the way it should (the form is missing elements/does not load properly), make corrections. • The fonts <i>should</i> be completely out of order- this will be remedied later with CSS. 	3	
Make corrections (if necessary)	<ul style="list-style-type: none"> • Open the HTML file in Visual Studio, and make sure that all the tags are closed, and that all the attributes, id's, etc. If they are not, make the requisite edits. • Reload the chrome page to see if the errors are fixed • Repeat this step as necessary, until the webpage works as intended. • Make sure to leave the HTML page open as a tab in Visual Studio. 	7	N.A

Create the stylesheet		Visual Studio	13 (in all)	
Create the CSS file	Open visual studio and press ⌘+n. Go to the 'web' section and select 'new CSS file'. Type 'style' into the name bar.		2	
Define the background color for the body	body {background-color:#006699; }		1	<pre>body { background-color:#006699; }</pre>
Define the heading style, in bold white with size 26	<ul style="list-style-type: none"> • h1{ • font-family: "Times New Roman", Times, serif; • font-size: 26px; • color: #ffffff; • font-style:bold; • } 		2	<pre>h1{ font-family: "Times New Roman", Times, serif; font-size: 26px; color: #ffffff; font-style:bold; }</pre>
Define the h2 style (for subtitles and help-text) with oblique test, size 10, and light weight.	<ul style="list-style-type: none"> • h2{ • font-family: "Times New Roman", Times, serif; • color: #ffffff; • font-style:oblique; • font-size: 10px; • font-weight:lighter; • } 		2	<pre>h2{ font-family: "Times New Roman", Times, serif; color: #ffffff; font-style:oblique; font-size: 10px; font-weight:lighter; }</pre>
Define the h3 style for form labels with size 13, colour white, and normal weighting	<ul style="list-style-type: none"> • h3{ • font-family: "Times New Roman", Times, serif; • font-size: 13px; • color: #ffffff; • font-weight: normal; • } 		2	<pre>h3{ font-family: "Times New Roman", Times, serif; font-size: 13px; color: #ffffff; font-weight: normal; }</pre>
Define the style for user inputs, with garamond font size 11	<ul style="list-style-type: none"> • input{ • font:'Garamond'; • font-size: 11px; • font-weight: normal; • } 		1	<pre>input{ font:'Garamond'; font-size: 11px; font-weight: normal; }</pre>
Define the textarea style (for the longer inputs), with the same settings as the input style.	<ul style="list-style-type: none"> • textarea{ • font:'Garamond'; • font-size: 11px; • font-weight: normal; • } 		1	<pre>textarea{ font:'Garamond'; font-size: 11px; font-weight: normal; }</pre>
Define the p style for invisible elements (URL and platforms)	p{ display:none;}		1	<pre>p{ display:none; }</pre>
Define the style for within the button, with a solid border, blue background and bold white text.	<ul style="list-style-type: none"> • button{ • border:ridge; • border-color:#ffffff; • background-color:#006699; • font-family: "Times New Roman", Times, serif; • font-style:bold; • font-size: 14px; 		2	<pre>button{ border:ridge; border-color:#ffffff; background-color:#006699; font-family: "Times New Roman", Times, serif; font-style:bold; font-size: 14px; }</pre>

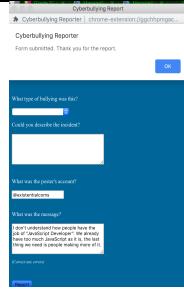
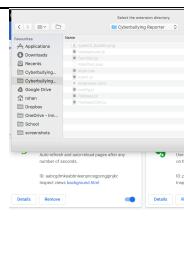
	<ul style="list-style-type: none"> • } 			
Save the CSS file	<ul style="list-style-type: none"> • Press ⌘+s • Navigate to 'cyberbullying report' in the dropdown menu and press the save button. • (Note: Remember to keep the CSS file open in Visual Studio after it is saved and complete) 		1	
Testing	<ul style="list-style-type: none"> • Navigate to the 'cyberbullying reporter' folder in Finder and open 'extension.html' in Chrome. • The page should have all the text appropriately formatted. • If not, go back to the CSS file and make sure that the content is according to the plan. • Make any corrections necessary (in the VS file) if they do not match, and save the file. • Reload the page in Chrome, and see if the errors are fixed. • Repeat this step as many times as necessary. 	Visual Studio Google Chrome	10	
Create the background.js script		Visual Studio	10 (in all)	
Create the JS file	Open visual studio and press ⌘+n. Go to the 'web' section and select 'new JS file'. Type 'background' into the name bar.		2	
Create the oninstalled function	<ul style="list-style-type: none"> • In the JS file, type the following code: • 'chrome.runtime.onInstalled.addListener(function() { • console.log ('I have awoken')/*This is just to make sure that the app has initialized properly.*\}); 		4	<pre>chrome.runtime.onInstalled.addListener(function() { console.log ('I have awoken')/*This is just to make sure that the app has initialized properly.*\});</pre>
Create the browseraction function, which injects a script into the current tab when the icon is clicked.	After the onInstalled function, insert the code <ul style="list-style-type: none"> • chrome.browserAction.onClicked.addListener(function() { • chrome.tabs.query({active: true, lastFocusedWindow: true}, function(tabs) { • var tab = tabs[0]; • chrome.tabs.executeScript(null,{file:'inject.js'}) • });}); 		3	<pre>chrome.browserAction.onClicked.addListener(function(tab) { chrome.tabs.query({active: true, lastFocusedWindow: true}, function(tabs) { var tab = tabs[0]; chrome.tabs.executeScript(null,{file:'inject.js'}); }); });</pre>
Save the background.js file	<ul style="list-style-type: none"> • Press ⌘+s • Navigate to 'cyberbullying report' in the dropdown menu and press the save button. • (Note: Remember to keep the JS file open in Visual Studio after it is saved and complete) 		1	
Create the inject.js file (to run on clicking the icon).		Visual Studio Chrome	23 (in all)	
Create the JS file	<ul style="list-style-type: none"> • Open visual studio and press ⌘+n. • Go to the 'web' section and select 'new JS file'. • Type 'inject' into the name bar. • Click the 'new' button. 		1	

Get URL and platform	<ul style="list-style-type: none"> In the inject.js document, type the code var current_URL=window.location.href; var current_Platform=window.location.hostname; 		1	<pre>var current_URL=window.location var current_Platform=window.lo</pre>
Write these values to storage.	After those previous lines, insert the code: <ul style="list-style-type: none"> chrome.storage.local.set({'current_URL':current_URL}); chrome.storage.local.set({'current_Platform':current_Platform}); 		1	<pre>chrome.storage.local chrome.storage.local</pre>
Initialize the message and account variables	Type the following: <ul style="list-style-type: none"> var current_Message=""; var current_Account=""; 		1	<pre>var current_Mess var current_Acc</pre>
Create the twitter extraction function	In a new line, insert the code <ul style="list-style-type: none"> if (current_Platform=='twitter.com') { current_Message=document.getElementsByClassName("TweetTextSize js-tweet-text tweet-text")[0].textContent; current_Account=document.getElementsByClassName('username u-dir u-textTruncate')[0].textContent; } 		3	<pre>if (current_Platform=='twitter.com') { current_Message=document.getElementsByClassName("TweetTextSize js-tweet-text tweet-text")[0].textContent; current_Account=document.getElementsByClassName('username u-dir u-textTruncate')[0].textContent; }</pre>
Create the facebook extraction function	In a new line, insert the code <ul style="list-style-type: none"> else if (current_Platform=="www.facebook.com") { current_Message= document.getElementsByClassName("_5pbx userContent _3576")[0].textContent; current_Account=document.getElementsByClassName("_2wmb")[0].textContent; } 		4	<pre>} else if (current_Platform=="www.facebook.com") { current_Message= document.getElementsByClassName("_5pbx userContent _3576")[0].textContent; current_Account=document.getElementsByClassName("_2wmb")[0].textContent; }</pre>
Create the instagram extraction function	In a new line, insert the code: <ul style="list-style-type: none"> else if (current_Platform=="www.instagram.com"){ current_Message="N.A- Image post"; current_Account=document.getElementsByClassName("F PmhX notranslate nJAzx")[0].innerHTML; } 		4	<pre>} else if (current_Platform=="www.instagram.com"){ current_Message="N.A- Image post"; current_Account=document.getElementsByClassName("F PmhX notranslate nJAzx")[0].innerHTML; }</pre>
Create the XKCD extractor function (for testing)	In a new line, insert the code <ul style="list-style-type: none"> else if (current_Platform=="xkcd.com"){ current_Message=document.getElementById('comic_hover_text').innerHTML;}; 		3	<pre>current_Account //xkcd is just a test value else if (current_Platform=="xkcd.com"){ current_Message=document.getElementById('comic_hover_text').innerHTML;};</pre>
Function to write the message and account to local storage	In a new line outside the 'if' loop, insert the code <ul style="list-style-type: none"> chrome.storage.local.set({'current_Message':current_Message}); chrome.storage.local.set({'current_Account':current_Account}); 		2	<pre>//writes current message chrome.storage.local chrome.storage.local</pre>
Create the code for opening the HTML file	In a new line, insert the code <ul style="list-style-type: none"> var extensionURL=chrome.runtime.getURL('/extension.html'); 		2	<pre>//opens form in new window var extensionURL=chrome.runtime.getURL('/extension.html');</pre>

	<ul style="list-style-type: none"> • window.open(extensionURL, '_blank', 'height=630,width=450, left=50, top=50'); 			
Save the inject.js file	<ul style="list-style-type: none"> • Press ⌘+s • Navigate to 'cyberbullying report' in the dropdown menu and press the save button. • (Note: Remember to keep the JS file open in Visual Studio after it is saved and complete) 		1	
Interlude: Testing and corrections	<ul style="list-style-type: none"> • Go to the Chrome extensions URL (chrome://extensions) • Turn on developer mode in the top right corner • Select the 'load unpacked' button. • Navigate to the 'Cyberbullying Reporter' folder in the pop-up • Click the 'open' button • The extension icon should appear as a button in the chrome toolbar • Go to a random webpage such as xkcd.com • Click the extension icon. The 'extension.html' file should open in a new popup. • If not, go back to the inject.js script and recheck for syntax errors such as unmatched parentheses. • Once any errors are fixed (in the inject.js file), update the extension using the small 'reload' button below the extension icon. • If it still doesn't work, press f12 to open the console and check for error messages. Make edits based on those. • Repeat this step as necessary, until it works as intended. 	Visual Studio Chrome	7	
Create the function.js file		Visual Studio	21 (in all)	
Prepare the function execution command	<ul style="list-style-type: none"> • At the top of the document, type the code <code>document.addEventListener('load',form_filler());</code> 		1	
Define the function 'form_filler'	<p>Below the above line, type the code <code>function form_filler(){}</code></p> <p>Place the cursor in between the curly parentheses</p>		1	
Write the code for retrieving the URL from storage, and filling it into the document:	<p>Type the following code:</p> <ul style="list-style-type: none"> • <code>chrome.storage.local.get(['current_URL'], function (item){</code> • <code>var current_URL=item.current_URL;</code> • <code>document.getElementById("URL").innerHTML=current_URL;</code> • <code>});</code> 		2	
Write the code for getting the platform and filling it into the document	<ul style="list-style-type: none"> • Select the above code, and copy it using ⌘+c. • Press the enter key, and paste it. Replace any instance of 'URL' in this snippet with 'Platform'. <p>The resultant code should read</p> <ul style="list-style-type: none"> • <code>chrome.storage.local.get(['current_Platform'], function (item){</code> • <code>var current_Platform=item.current_Platform;</code> 		1	

	<ul style="list-style-type: none"> • document.getElementById("platform").innerHTML=curr ent_Platform; • }); 		
Write the code for getting the account and filling it into the form	Press the enter key and paste the same snippet again. Replace any instance of URL in this snippet with 'Account'. The resulting code should look like: <ul style="list-style-type: none"> • chrome.storage.local.get(['current_Account], function (item){ • var current_Account=item.current_Account; • document.getElementById("account").innerHTML= current_Account; • }); 	1	<pre>chrome.storage.local.get(['current_Account], function (item){ var current_Account=item.current_Account; document.getElementById("account").innerHTML= current_Account; });</pre>
Write the code for getting the message and filling it into the form	Press the enter key and paste the same snippet again. Replace any instance of URL in this snippet with 'Message'. The resulting code should look like: <ul style="list-style-type: none"> • chrome.storage.local.get(['current_Message'], function (item){ • var current_Message=item.current_Message; • document.getElementById("message").value=current_Mess age; • }); 	1	<pre>chrome.storage.local.get(['current_Message'], function (item){ var current_Message=item.current_Message; document.getElementById("message").value=current_Mess age; });</pre>
Insert the code to execute the 'firebase_writer' function when the submit button is clicked.	document.getElementById("submit").onclick=function() {firebase_writer();};	1	<pre>document.getElementById("submit").onclick=function() { firebase_writer(); };</pre>
Define the 'firebase_writer' function	Type the code “function firebase_writer(){ };” Place the cursor between the curly brackets	1	<pre>function firebase_writer(){ };</pre>
Get the values entered in the form as local variables	<ul style="list-style-type: none"> • var name=document.getElementById('name').value; • var description = document.getElementById('description').value; • var account= document.getElementById('account').value; • var message=document.getElementById('message').value; 	2	<pre>//gets the data to be push var name=document.getElementById('name').value; var description = document.getElementById('description').value; var account= document.getElementById('account').value; var message=document.getElementById('message').value;</pre>
Initialize the database, and prepare the path to write the data	Type the following code: <ul style="list-style-type: none"> • var database.firebaseio.database(); • var ref= new Firebase("https://file-2e88a.firebaseio.com"); • var file=ref.child("Reports"); 	1	<pre>//initialize ev var database=fi var ref= new Fi var file=ref.ch</pre>
Get the URL and platform from the HTML document as variables	Type the following code: <ul style="list-style-type: none"> • var URL= document.getElementById('URL').innerHTML; • var platform= document.getElementById('platform').innerHTML; 	2	N.A (changed slightl

Get the text content of the selected type of bullying.	<p>Type the following code:</p> <ul style="list-style-type: none"> • var x = document.getElementById("type").selectedIndex; • var y = document.getElementById("type").options; • var selection=y[x].text; 		2	<pre>//gets the type of b var x = document.get var y = document.get var selection=y[x].t</pre>
Push all the data to the firebase database, and inform the user of the submission	<ul style="list-style-type: none"> • file.push({ • Name: name, Description: description, URL: URL, Platform: platform, Account: account, Message: message, Timestamp: Date(), Type: selection • }); • window.alert('Form submitted. Thank you for the report.'); 		4	<pre>//push the data to firebas file.push({ Name: name, Description: descripti URL: URL, Platform: platform, Account: account, Message: message, Timestamp: Date(), Type: selection }); window.alert('Form sub</pre>
Save the function.js file.	<ul style="list-style-type: none"> • Press ⌘+s • Navigate to 'cyberbullying report' in the dropdown menu and press the save button. • (Note: Remember to keep the JS file open in Visual Studio after it is saved and complete) 		1	
Uploads and testing		Chrome Visual Studio (maybe)	23 (in all)	
Upload the final file	<ul style="list-style-type: none"> • Open Google Chrome • Go to the page chrome://extensions. • Turn on developer mode in the top-right corner if it isn't on already • Select the 'load unpacked' option. • Navigate to the 'cyberbullying reporter' folder in the popup • Click the blue 'open' button 		3	
Initiate testing	<ul style="list-style-type: none"> • In a new tab in chrome, visit the following URL: https://twitter.com/existentialcoms/status/1060710147671044096 (any twitter URL will work, really). • Click the extension logo (the blue hand) in the chrome extension bar. 		1	
Test the data retrieval for twitter	<ul style="list-style-type: none"> • The webpage extension.html should open in a popup. The 'account' field should be filled with @existentialcoms, and the 'message' with the text 'I don't understand how people have the job of "JavaScript Developer". We already have too much JavaScript as it is, the last thing we need is people making more of it.' • If not, click f12 to open the console. Check for error messages such as 'syntax error' or 'unexpected identifier'. Based on these error messages, return to the function.js file and make any syntax/typo corrections. 		7	

	<ul style="list-style-type: none"> • Reload the extension using the small reload button below the extension pane (in chrome://extensions). • Return to the URL, try again, and repeat as necessary. 		
Test the data writing	<ul style="list-style-type: none"> • Once the data retrieval function works, visit the same URL (https://twitter.com/existentialcoms/status/1060710147671044096). • Click the button to launch the extension. • Press f12 to open the console. Make sure no error messages occur. • Press the 'report' button in the popup. Click the 'OK' button in the dialog box, and close the window. 	7	
Check if firebase received the data	<ul style="list-style-type: none"> • Go to the firebase database URL (https://file-2e88a.firebaseio.com/) • Expand the latest report (which will be the last item in the list). <p>The data should look like this:</p> <pre> • { • "Account" : "@existentialcoms", • "Description" : "", • "Message" : "I don't understand how people have the job of \"JavaScript Developer\". We already have too much JavaScript as it is, the last thing we need is people making more of it.", • "Name" : "", • "Platform" : "twitter.com", • "Timestamp" : (Whatever the time was when the report was sent) • "Type" : "", • "URL" : "https://twitter.com/existentialcoms/status/106071014767104409" } </pre> <p>If not, return to the firebase_writer function in function.js and recheck the syntax, parentheses, etc.</p> <p>Repeat as necessary</p>	5	
Hofstadter's Law	This is not a step as such, just an additional allowance for time in accordance with the principle that it always takes longer than you expect.	60	N.A
Get the extension as a CRX file	<ul style="list-style-type: none"> • Open Google Chrome • Go to the page chrome://extensions. • Turn on developer mode in the top-right corner if it isn't on already • Select the 'pack extension' option. • Navigate to the 'cyberbullying reporter' folder in the popup 	Chrome developer mode	

	Click the blue 'open' button			
Upload to GitHub	<p>Go to GitHub.com</p> <p>Click the + icon and select 'new repository'.</p> <p>Type 'cyberbullying reporter' as the repository name, and click 'create repository'</p> <p>Click the link that says 'upload existing files'. Drag and drop first the 'cyberbullying reporter' folder and then the cyberbullying reporter.crx file.</p> <p>Once the files are done uploading, click the 'create repository' button.</p> <p>Click the 'add ReadME' button</p>	GitHub repositories	5	
Create GitHub readme.md	<p>Type the following text in the readme:</p> <hr/> <pre>##Instructions</pre> <hr/> <p>First, download the CRX file from the repository. Then go to the URL chrome://extensions in Google Chrome.</p> <p>Make sure developer mode is switched on (the small switch in the top right corner).</p> <p>Drag and drop the CRX file from your downloads into the chrome extensions window.</p> <p>Click the 'Add Extension' button in the dialogue box.</p> <p>A small blue hand should appear in your Chrome Extensions Bar.</p> <p>If you see cyberbullying on any site, simply click the blue hand.</p> <p>A web form will pop up which you can fill in. The URL and timestamp will be automatically logged.</p> <p>Depending on the site you are reporting from, the message and account might be already filled. If so, simply check over them and correct any errors.</p> <p>Once the form is filled, click report and click 'OK' on the dialogue box. Then close the window and return to whatever you were doing.</p>	Inbuilt markdown editor		

Total Time (A lot of the code can be copy-pasted from here, rather than painstakingly typed): 157 mins=2.5hrs

Total Time (accounting for Hofstadter's law): 217 mins=3.5hrs

Large-size screenshots can be found in Appendix C

Changes made:

Step	Changes Made	Justification
Creating the manifest	At the end of the manifest (just before declaring the manifest number) I inserted a line "content_security_policy": "script-src 'self' https://cdn.firebaseio.com https://*.firebaseio.com https://www.googleapis.com; object-src 'self'; connect-src 'self' wss://*.firebaseio.com;";	When I attempted to test the extension, it sometimes failed to submit to firebase, and showed an error message regarding chrome's 'Content Security Policy'. I googled the error message, and found a blog post (cited in appendix) which said that this line must be added to the manifest to allow firebase scripts to load. When I edited the manifest, and tested it again, the extension worked perfectly.
Create the 'description' textarea	Specified the size (rows and columns) as attributes, giving the textarea a fixed size. (5 rows by 35 columns)	The default size was too small to be valid for most messages, and required users to inconveniently scroll. Thus, I had to adjust the length and width to be more appropriate.
Create the 'message' textarea	" "(As above)	" "(As above)
[General]	Inserted comments into JS files, which explain in layman's language what various blocks of code do. Before each function or block of code, I inserted a brief sentence explaining what it does, such as //This function gets the URL and writes it to local storage	Adding annotations is considered good practice in programming. Since I am making this for an organization, they might need to edit/maintain it. In that case, it would be better if they can understand the code without having to parse every line of JavaScript, especially since many might be unfamiliar with Chrome APIs and thus be a little confused.
Create the MCQ	Added an additional 'empty' option within the <select> tag: <option value=""></option>	Without this option, the default was set to 'vulgar/rude language'. This might bias our data in case of users who leave it set to the default. With this addition, at least we have no data rather than misleading data.
Creating the instagram content extractor	Wrapped the 'get account' code in an if statement: if (typeof document.getElementsByClassName("FPmhX notranslate nJAzx")[0] != "undefined").	On instagram pages which do not display a specific post, there is no element with the class "FPmhX notranslate nJAzx". Thus, on this sort of page my code threw an error (cannot get innerHTML of undefined) and failed to execute. To remedy this I had to use an 'if' to make sure such an element existed, and was not undefined.

Screenshot of the form with data extracted from twitter post

The screenshot displays two windows side-by-side. On the left is the 'Cyberbullying Report' extension interface, which includes fields for 'Name', 'What type of bullying was this?', 'Could you describe the incident?', 'What was the poster's account?', and 'What was the message?'. The right window shows a Twitter post by user @tomscott. The tweet content is:

♪ If there's something strange
♪ In your neighbourhood
♪ Who you gonna not call because they
don't have an online booking service
and making phone calls to strangers is a
weird and unnatural anxiety-inducing
thing to many people born after the mid-
eighties or later ♪

The tweet was posted at 1:34 PM - 30 Sep 2018, has 361 Retweets, 1,566 Likes, and 1.6K interactions. The extension interface shows some of the tweet content being processed or extracted.

Screenshot of the record in the database

```
-LRSNGDMbLrmHzCvXAOh
  .... Account: "@tomscott"
  .... Description: ""
  .... Message: "♪ If there's something strange\n♪ In your neigbh..."
  .... Name: ""
  .... Platform: "twitter.com"
  .... Timestamp: "Fri Nov 16 2018 20:36:00 GMT+0400 (Gulf Standar..."
  .... Type: ""
  .... URL: "https://twitter.com/tomscott/status/10464986661..."
```

The product is intended to provide students at RWA with an easy and safe way to report cyberbullying to the Anti Bullying Committee; preventing victims from suffering in silence and watchers from remaining bystanders. By doing this it should also help the Anti Bullying Committee become more aware of cyberbullying, since records are easier to keep and obtain, and thus in the long term let us combat it more effectively, such as identifying and responding to accounts with a habit of cyberbullying. [develop]

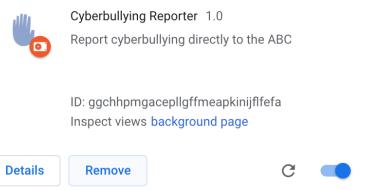
The GitHub repository from where my project can be downloaded (both as a folder and as a chrome extension file): <https://github.com/Rohan-Goyal/cyberbullying-reporter>

Criterion D

Product Testing

Question	Test	Criterion	Result	Pass/Fail
Is the project complete?	Ask the client	If the client agrees it is complete	[See client survey]	Pass
Does it fulfil its aim of combatting cyberbullying?	Ask the client	If the client agrees it is successful	[See client survey]	Pass
Is the UX intuitive and accessible?	Ask 3 peers	If they agree that it is intuitive	<ul style="list-style-type: none"> • Kshitaij: Yes • Adi: Yes • Vandith: Yes 	Pass
Is the database easy to access and use?	Ask the client	If they believe it is easy to use	[See client survey]	Pass

Specification Testing

Specification	Test	Criterion	Results	Pass/Fail
Retrieving the evidence in the form of URLs and/or text should be automated for major social media sites (Twitter, Facebook, Instagram).	The URL is sent to the database, even if it is not entered by the user.	Screenshot of URL in database	<pre>-LRHnq2CkOBY2s5fB6C4 Account: "@existentialcoms" Description: "" Message: "I don't understand how people have the job of \\"... Name: "" Platform: "twitter.com" Timestamp: "Wed Nov 14 2018 19:20:39 GMT+0400 (Gulf Standar... Type: "" URL: "https://twitter.com/existentialcoms/status/1060..."</pre>	Pass
Report data must be stored in SQL or some web database, so that it can be logged or analyzed	Screenshot of the database.	The database contains sample reports as data	<pre>-LRHnq2CkOBY2s5fB6C4 Account: "@existentialcoms" Description: "" Message: "I don't understand how people have the job of \\"... Name: "" Platform: "twitter.com" Timestamp: "Wed Nov 14 2018 19:20:39 GMT+0400 (Gulf Standar... Type: "" URL: "https://twitter.com/existentialcoms/status/1060..."</pre>	Pass
The product should be able to run perfectly as a chrome extension	Screenshot of the extension pane	No errors when uploading it	 <p>Cyberbullying Reporter 1.0 Report cyberbullying directly to the ABC ID: ggchhpmgaceplgfmeapkinijffefa Inspect views background page Details Remove C</p>	Pass
It should use efficient coding and minimal media to load quickly.	Screenshot of the extension folder, and media contained	Less than 3 images are used	 <p>style.css extension.html background.js config.js firebase.js firebaseCDN.js function.js inject.js Manifest.json speech_bubble.png</p>	Pass (only 1 image used)
It should be equipped to deal with all the various types of cyberbullying: harassment, rumor-mongering, flaming, etc.	Ask the client what they think	The client agrees that it meets this requirement	[See client survey]	Pass
The app should be easy enough to use that only a few	Ask 3 peers	3/3 peers feel that it is intuitive	<ul style="list-style-type: none"> • Kshitaij: Yes • Adi: Yes 	Pass

sentences of instruction are necessary; ideally, it should be completely intuitive.			<ul style="list-style-type: none"> • Vandith: Yes 	
The form fields should be unambiguously labelled	Ask 3 peers	3/3 should agree that the fields are clearly labelled	<ul style="list-style-type: none"> • Kshitaij: Yes • Adi: Yes • Vandith: Yes 	Pass
Anything relating to evidence of the incident should be automatically filled	Screenshot of report from Facebook post.	Message text and account fields are auto-filled	<p>What was the poster's account? @tomscott</p> <p>What was the message? It's been a while since I've just taken a camera to a place. This feels like a good video.</p> <p>(Correct any errors)</p>	Pass
There must be an option for anonymous reports	Submission of sample nameless report.	Report shows up in database with empty 'name' field	<pre>-LRHnq2CkOBY2s5fB6C4 └ Account: "@existentialcoms" └ Description: "" └ Message: "I don't understand how people have the job of \\"... └ Name: "" └ Platform: "twitter.com" └ Timestamp: "Wed Nov 14 2018 19:20:39 GMT+0400 (Gulf Standard Time)" └ Type: "" └ URL: "https://twitter.com/existentialcoms/status/1060..."</pre>	Pass
The form should have a minimalist design, with only a few colors/ styles	Ask 3 peers	3/3 agree that it has a minimalist design	<ul style="list-style-type: none"> • Kshitaij: Yes • Adi: Yes • Vandith: Yes 	Pass
The form should have a calming blue-white color scheme to put users at ease	Screenshot the web form	The dominant colors are blue and white		Pass
The icon should be small enough for the extension bar	Screenshot of extension bar	The icon (blue hand) is the same size as other extension icons,		Pass
It must have a bold color to stand out.	Ask 3 peers if it stands out	2/3 agree that it does	<ul style="list-style-type: none"> • Kshitaij: No • Aditya: Yes • Vandith: Kind of 	Fail (borderline)
It should give a good idea as to the purpose of the app	Ask the client	The client agrees	[See client survey]	Pass
The report data should be easily accessible by the admin users, and not require complex retrieval.	Ask the client	The client agrees	[See client survey]	Pass
A way of notifying admins when complaints are made	Ask the client	The client agrees	[See client survey]	Fail

Self-Testing Analysis:

The product passed nearly all the specifications and I am satisfied that the product accomplished its purpose.

Almost all of the technical specifications were passed, including the automated functions (which are very much the USP of this solution). This added functionality to the extension and makes it far easier to use. The information is also successfully stored in a database, which means that it can be acted on by the ABC, and the product has thus fulfilled its requirements. The basic product testing also yields positive results, since the product works as intended and appears to be bug-free; it is also simple enough that I can actually believe that to be true.

The product also successfully passed most of the peer feedback-based specifications, on both aesthetic and ergonomic fronts. My peers agreed that the automated functions greatly improved the User Experience, and that the UX was quite intuitive and user-friendly, which my client stressed was extremely important. This ensures that the product is quite likely to be used by my audience, and thus generate more relevant data. They also agreed that the form fields were unambiguously labelled and easy-to-understand, which further improves the UX.

My client was also satisfied with the product; she agreed that it was well-equipped to deal with all the various types of cyberbullying rather than with just a few, and also that the database structure (which, in fairness, was designed by Google Firebase) makes it very easy to monitor and retrieve reports.

The main specification which it failed (database notifications) is a failure on the administrator side, i.e. the database. This can be excused as my client and I agreed that working with the database was essentially a separate project which lies outside the scope of this one; my primary aim was to provide students with a way to report cyberbullying which has been accomplished. Additionally, working with the database in such a way would have been complicated, especially before the system had formally been introduced. It would have required knowing who was going to be involved, and individual permissions for their devices, which can only be done once the project is established. The other failure of the project is that the logo is somewhat unclear, which can be easily solved by making it a darker shade. Nonetheless, the logo is at best a minor part of the project; the key elements all work well and accomplish their purpose.

Peer/Audience Feedback (Since my audience is almost entirely secondary school students, the two groups are essentially equivalent).

Note: This is only a summary. Full records can be found in the appendix D as a table.

Question	Summary
Is the frontend UX easy to understand?	Yes
Is the database easy to use?	Yes
Do the automated functions improve the ergonomics?	Significantly
Should the aesthetics be more or less elaborate? (Or is it fine as it is)	Somewhat more elaborate, especially in the database
Rate the color scheme out of 10	7.67
Are the form fields clearly labelled	Yes
Does the icon stand out clearly?	Partially; it is visible but could be much more clear.
Does it convey the purpose of the extension?	Slightly, but not at all obviously.
How likely (%) are you to use this?	75.5% (on average)
Overall, give it a rating out of 10	7.8
Can you suggest any improvements?	<ul style="list-style-type: none">• More detailed form• Database Aesthetics• More visible icon• Center-aligned form• Larger, more professional font/ visuals• Screenshot of offensive content as well as URL and text.

Analysis:

The feedback on the product from my peers was quite satisfactory although not 100% positive. All of them agreed that the UX was easy to understand, both on the front and back end. They also agreed that the ergonomics and technical functions of the project are up to the mark. Most importantly, all of them agreed that they were more than 50% likely to use the app to report cyberbullying, except for one person- and their reasoning was not to do with the app itself but with the fact that they don't usually notice cyberbullying. Compared to the normal stats for reporting cyberbullying, where going unreported is the norm, that is a significant improvement indeed. On average it seems people are 75.5% likely to gain significant utility out of this, which I believe qualifies as a complete success- ultimately, the test of a solution is whether or not it is actually useful to people.

My classmates also uniformly agreed that the automated functions such as the URL greatly improved the ergonomics of the site; this is a feature that I also put a lot of importance on (hence my constant bringing it up)

The aesthetics of the product were somewhat below par, but thankfully not enough to detract from its usability or success. Most of my classmates felt that they had potential to be more detailed, in particular suggesting that I use a larger and more professional-looking font.

The main suggestion given by my peers was to improve the aesthetics of the database (which is outside the scope of this project) and of the icon. They noted that the icon is not that clear (partly because of the way I have customized my browser background and colours) and that it should be made brighter and more saturated. They also felt that while the colour scheme was quite good, it still had room for becoming a little more attractive and vibrant. Another feature they suggested. However, I worry that making it too detailed might lower the usage rate; the client is also of the opinion that the factors mentioned in the form are sufficient.

Client Feedback (proof in appendix D)

Question	Answer
Is the product complete?	The product is complete. Very thorough work.
Does the product fulfil your requirements?	It does fulfil our requirements as the ABC initiative to eradicate all forms of cyberbullying.
Are the types of bullying mentioned sufficiently extensive? (Mentioned are: Vulgar/rude language, Gossiping, Harassment, Impersonation, revealing private info, Other/Don't Know)	They are mentioned in the drop-down menu and briefly explained as needed.
In general, comment on the success of the product.	The product is successful, innovative and user-friendly.
Rank the following aspects of the product: aesthetics, usability, efficiency.	It is easy to use, simple to navigate, language is relatable. (Efficiency, Usability, Aesthetics)
Will the product make it easier for the ABC to combat cyberbullying?	it is will be pioneered first among ABC members to check efficiency. Overall, it is very helpful and needed to stop cyberbullying incidents.
Does the form collect sufficient data regarding cyberbullying incidents? [On a scale of 1-10, 1 being 'not at all' and 10 being 'very']	It does collect relevant data efficiently. Scale: 10
Is the data the form collects relevant? [On a scale of 1-10]	The data supports the aim of the application and is relevant and appropriate to the ABC's aim. Scale: 10
Is the database/backend easy to understand and use? [On a scale of 1-10]	Yes. 10/10
Is the frontend UI easy to use? [On a scale of 1-10]	Yes. 10/10
How appropriate is the interface (logo, page layout, structure, etc.) to the purpose/theme?	The logo is completely appropriate. The page layout is appropriately formal, but it could communicate the purpose more clearly (such as with an ABC logo).
Suggest any improvements which could be made	<ul style="list-style-type: none">• Improve the aesthetics of the database somewhat.• Implement a feature for the ABC to open a dialogue• Have notifications for administrators when a report is added.

Analysis

My client seems quite pleased with the product; far more than I anticipated based on the previous testing. She agreed that the data collected by the form was completely sufficient and relevant (rating both aspects 10/10), and that the product was complete to an acceptable degree (calling it ‘very thorough’). She also rated the UX 10/10 both for the form and for the database, which agrees with my previous results and lends strength to them. She also expressed the opinion that the types of cyberbullying for classification mentioned in the dropdown question were sufficiently extensive, and also well explained. Overall, my client agreed that the app was user-friendly, effective, and thus a successful project; she was even kind enough to call it innovative and efficient.

Although she rated aesthetics at the bottom, she implied that it was nonetheless successful, since she rated the UI 10/10- this shows that the aesthetics were not bad enough to hinder the success of the app. The main improvements she suggested were to develop the database and backend system, which I will likely be doing as a separate project. She suggested that there should be some way for the ABC to communicate with the offender, as well as a way to notify admins (i.e. members of the ABC) when a report was made and sent, which would require buy-in from the whole committee. My client’s final decision for the project was to first give it a limited release solely to ABC members who could be relied upon to use it appropriately. This is one I wholeheartedly agree with, as it will allow me to fix any systemic flaws before they can become too much of a problem; it will also allow me to generate more feedback data.

Improvements:

A way to get email addresses (of reporters or offenders) and open a dialogue: My client suggested that to have a serious impact, the service should allow the ABC to open a dialogue with a cyberbully. She suggested that it begin with an automated email which informs the bully of the event, and politely asks them to apologize to the victim. It could later evolve into a full dialogue between the victim and bully, mediated by the ABC. This feature would make the app far more effective, since it raises the possibility of the ABC actively participating, and potentially ‘rehabilitating’ a bully. This feature would require enlisting an automated email service (which can be done using node.js), as well as the ability to extract email addresses of users (which would be quite difficult, and require a developer account for facebook, twitter, etc.). Since the ABC can communicate with the offenders account via conventional and manual means, this pitfall is not as significant; nonetheless this is a feature which I believe should be added. (it was also explicitly suggested by my client).

One of my peers suggested that the offensive content might be deleted by the offender or taken down by the site, in which case the matter would be out of the ABC’s hands. He suggested that to supplement this the app should also get a screenshot of the offending content and send that to the ABC as well. This functionality would also make it possible for people to report offensive images, whose content cannot be easily translated to text (such as Instagram posts). This would require implementing a firebase storage system as well as a database on the backend. On the frontend, HTML forms have an option to insert files and chrome also has a function to capture the visible content of a tab; together I could probably use these features to implement this idea, although I would need to research them in detail first.

Remove the dialogue box: As it stands, the dialogue box at the end of submission simply informs the user that the form has been submitted, which is essentially useless. Worse, it adds a pointless click to the user’s experience since they have to close the dialogue box. A more effective method would be to have a confirmation dialogue box, which allows the user to cancel his submission. Additionally, the form should ideally clean up after itself; it should automatically close after the user submits a report since they have no further use for it. These two functions can be implemented using the window.confirm() and window.close() methods in JavaScript. Having the window auto-close essentially removes a click, streamlining the UX. This can be accomplished by adding a window.close() command to the JS function just after the form data is submitted. The confirmation can be accomplished by placing the entire file.push command within an “if window.confirm==true {}” loop, and allows users more control over what they submit.

Logo: The logo, as it stands, is a very pale blue which is easy to miss. Many of my peers noted this, and suggested that I improve it. Thus, I intend to make it a slightly darker shade and increase the saturation of the image (using Gimp), and re-upload the extension. Depending on later feedback, I might simply change to a different style of logo, one taken from a professional site (with credit and permission obviously).

Font: The font of the questions is currently too small to read, and it also looks somewhat unprofessional (since it is an uncommon font). In order to improve the ergonomics, I should increase the size of the font somewhat by editing the appropriate CSS style. I should probably increase it from 13px to 15 or 16px, to make it easier to see.

Include more automation: Currently the automated features only work on facebook, twitter, and Instagram. However (based on my research), I know that it is also common on other sites. Thus I should eventually implement this feature for other social media sites; YouTube would be a good start. This expansion is fairly simple, since it will work on the same principles as the other functions; it would mostly require studying the HTML source to find out how YouTube encodes comments.

The following are essentially part of a separate project, which would supplement this one.

Queries for the database: This is a feature I intended to implement, but abandoned when I realized that I had my hands full with designing the frontend. Instituting queries would allow more focused searches by administrators. It would allow them to search for specific messages and reports to be dealt with, and allow them to do things such as search for swearwords (using lists or regexes). This would greatly improve the effectiveness of the database, and will become a necessity should the extension become as widespread as I hope it does. Firebase does have options for querying specific nodes. It also has the option to retrieve the data as a JavaScript object or as plain text, which can then be searched and queried based on text content, etc. Even if that proves impracticable, the structure of my database and code means that it can be refactored easily to accommodate any major changes to the database structure which might become necessary.

Data analytics: In order to truly be effective in preventing rather than just combatting cyberbullying, the extension should have some analytical and predictive power. For instance, it should be able to analyze basic text patterns, as well as graph trends in types of cyberbullying, and prevalence of it on various platforms. The ability to analyze and ferret out these trends would both be an interesting technical problem, and an invaluable resource for the ABC to understand cyberbullying better. This sort of analytics would need to be done in Python somehow; since firebase has some support for Python software this should be possible.

Notifications: The database or backend website should have a way to notify admins when a report is made, so that they can respond to it efficiently and effectively. Otherwise, reports may go unnoticed and undealt with for too long. Firebase has cloud functions which connect with their database, which allows us to call certain functions when data is written. This can be done to create web notifications, as well as push notifications for some devices and websites.

A more aesthetically pleasing UI for the database: Many of my peers, and my client, feel that there should be a more aesthetically pleasing way to access the database. My client suggested that it be an entire website with inspirational quotes, and she suggested that it also have resources and information on cyberbullying.

Final Appraisal

Overall, I believe that the product was quite successful. The app is quite impressive as a technical feat, and its technical features make it far more effective. By creating the product as an extension rather than simply as a webpage, it is ever-present in most people's browsers and thus far easier to use.

Analysis of Impact:

The product has not yet been introduced, and so it is somewhat difficult to accurately assess or analyze its impact. Nonetheless, based on feedback from my client and from my audience/peers, I can estimate the impact it will have if/when it is introduced. I believe that the extension would have a very significant impact and become quite widely adopted. On average, my classmates estimated that they had a 75% chance of using it if they experienced or saw cyberbullying. Thus, I believe that the app is highly likely to decrease the prevalence of cyber bullying in the school.

My client stated that she would first introduce this among ABC members. I believe that it will have