iWitness: A Police Interactions Monitoring Application

Final Report for CISC499 - Winter 2022

Anthony Rios, Joshua Chai-Tang, and Aidan Turnbull
Supervisor: Dr. Catherine Stinson

1. Abstract

iWitness is a collaboration with a Canadian community of color that experiences an adverse relationship with their local law enforcement. They requested an application that would enable users to document, archive, and visualize police interactions within their community. With the assistance of iWitness racist, abusive, as well as overall inappropriate behavior can be recorded and broadcasted so the community can uncover patterns while also accumulate evidence against problematic officers. Following a year of consultations, three undergraduate computing students designed and built a functional web application incorporating the group's specifications. Once deployed, the application will be maintained by our community partners and hopefully will contribute to a safer policing environment.

2. Introduction & Background

This project is a collaboration with a community of colour that routinely faces police harassment, including: needless stops, suspicious tailings, and unwanted visits in their own home. Unfortunately, this community's relationship with the police is not an outlier for Canadians of colour. In Toronto, the most multicultural city in the world, one third of all arrests are black despite black Torontontions only accounting for eight percent of the population [1A]. Similar unjust treatment across Canada has contributed to one-in-five black Canadians having no trust in their police [2A].

With the introduction of smartphones into the mainstream, videos of violent police interactions have entered the public discourse and opened the public's eyes to how law enforcement treats some citizens; particularly citizens of colour. Through video evidence police can no longer rely on their moral authority and "he said, she said " arguments to evade accountability. This has been seen in Chicago where the implementation of body cameras has increased the likelihood of civilian complaints resulting in officer discipline from 2.1 to 64 percent [3A]. Despite these positive indications, relying on body cameras is not enough when the majority of footage is never released to the public [4A]. It is usually bystander videos that make the biggest difference when it comes to triggering police accountability. It is for this reason our community partners hope to harness the power of crowdsourcing with this application to collect interaction data and catalyze change in their neighborhoods.

3. Impact Statement

Building a map-based documenting application will inevitably introduce a lot of information into the public domain. Although there are many benefits of real time information, it's important to consider the different ways this application could impact both the community and the police.

Providing the public with a method of documenting police interactions will eventually create a robust dataset. Increased sampling will ideally translate into accurate data and uncover potential policing trends. Moreover, having this information public will provide activists, city officials, as well as regular citizens unfiltered access to how civil servants are treating their community. Knowing the public will be able to directly witness their actions should remind police they are accountable and hopefully improve their behavior. Furthermore, if the police continue to behave unacceptably the community will have an archive of information to demonstrate how the police are over-resourced and unjustly focusing on particular neighborhoods.

Any technology that increases the public's access to policing knowledge runs the risk of being used nefariously. Actual criminals could use this application to track police patterns and movements to time crimes and avoid being caught. Furthermore, in order to allow all members to participate there cannot be any barriers to posting an interaction. Unfortunately, this creates an ability for internet trolls to spam the system and pollute the dataset. Another consideration is the sensitivity of the information being uploaded. Photos/videos of individuals could lead to doxxing or increased targeting by police who are unhappy about the documentation. Finally, depending on how the application is used (documenting every interaction or just negative ones) the dataset could become biased and potentially hinder the ability of the community and police to reconcile with each other.

Since the community group requested this application the positive impacts of collecting policing information outweigh the potential misuses. Furthermore, with continued development security and validity issues can be contained by implementing new features. Ultimately, providing access to information is almost always a positive. It allows individuals to learn the facts, make their own assessments, and take action if necessary.

4. Requirements

The following section is a copy of the specifications document provided to us. In most regards our project was based on these requirements exactly. The second column reflects the fulfillment status of each requirement in our final submission of the app.

Requirement	**Status**	**Notes**
Functional requirements		
Accessible, secure, and easy to use by public	Incomplete	The app isn't available to the public, but should a dedicated server be established it will be easy to use. 'Secure' is not satisfied, as the site's cybersecurity features need improvement before a public launch makes sense.
Allow user to document incidents	Complete	
Allow user to visualize incidents	Complete	
Allow user to choose the type of incident	Complete	The list of incident types is currently very short, but it could be easily expanded.
Allow user to choose privacy level	Incomplete	
Visualizing the data in the form of an interactive map	Completed	
Store data securely in the database	Incomplete	Passwords are hashed, but otherwise nothing is encrypted.
Landing page		
Purpose is to record an incident	Complete	
Drop a pin for an incident (use location data if allowed)	Complete	
Start recording video/audio	Complete	
Map of incidents	Complete	
My history	Complete	
Starting to record video/take picture should be easy	Complete	

Pop-up warning when posting video/audio about identification risks	Incomplete	
Adding pin to map should be quick and easy	Complete	
Quick access to app such as favorite shortcut from lock screen	Incomplete	Not necessary for a web app. Users may use browser bookmarks.
Add/Edit Incident page		
Most common options visible, click to open more options	Complete	
Allow users to tick off multiple 'incident types'	Complete	The list of incident types is currently very short, but it could be easily expanded.
Location and time (grab automatically, but also allow editing)	Complete	
License plate (of police vehicle), ID officer (name, badge, photo, description)	Complete	
Private description (not for display on map)	Complete	Private descriptions for your reports are shown on the My History page.
Attach video, audio or photo documentation	Complete	
Option to delay a pin showing on the map (for safety reasons)	Incomplete	Date/time of report uploads are tracked internally, but this information isn't used for anything.
Delete from device (but keep on server) and save on device (but do not send to server)	Incomplete	
Delete incident from server (opens a request form)	Incomplete	Currently incidents must be deleted manually by developers using SQLite commands.
Incident Map page		

Click/hover over a pin to see incident details	Complete	
Visualization of some basic stats about incidents in the area	Incomplete	
Choice of cities	Incomplete	
Should have some sensible neighborhood boundaries	Incomplete	Leaflet does support the kind of map overlays needed for this feature, but the boundaries would have to be added manually using external neighborhood maps as references.
Zoom in and out	Complete	
My History page		
List of incidents recorded from this device/user, allow to view and edit them	Partially Complete	The page does show a list of incidents from the current user, but there aren't any options to edit them.
Option to have a username attached to pins, or to pin anonymously	Incomplete	All pins are automatically anonymous at this time.
Security settings (default to removing metadata from reports, but have the option to turn it on)	Incomplete	
Options to display all info or to only display the pin (date, location and incident type)	Incomplete	This page does not contain a map, only a list.
About Page		
Resources on police accountability	Partially Complete	Page is there, just need to add the appropriate information. Guidance required as to what resources should be included.
Link to safety / legal notes	Partially Complete	Page is there, just need to add the appropriate information. Guidance required as to what notes should be linked.

Short intro provided by community group	Partially Complete	Page is there, just need to add the appropriate information. Guidance required as to what this intro contains.

5. Methodology & Implementation

5.1. Features

This project is implemented as a website based application. Thus it is accessible from any standard web browser on any device, both mobile and desktop. All of the site's features are mobile compatible, including the map. However the user interface is primarily designed for desktop resolutions, and has not been fully tested with alternative device sizes.

The app's core feature is the incident map: a comprehensive street map which displays pins at the locations of user reported incidents. This map is prominently displayed on the site's home page (see figure 1). Each pin can be interacted with to reveal details about the incident based on information the user provided. This includes a textual description of the incident, the date and time that it occured, details about the officer involved, and links to attached photos and/or videos, if any (see figure 2).

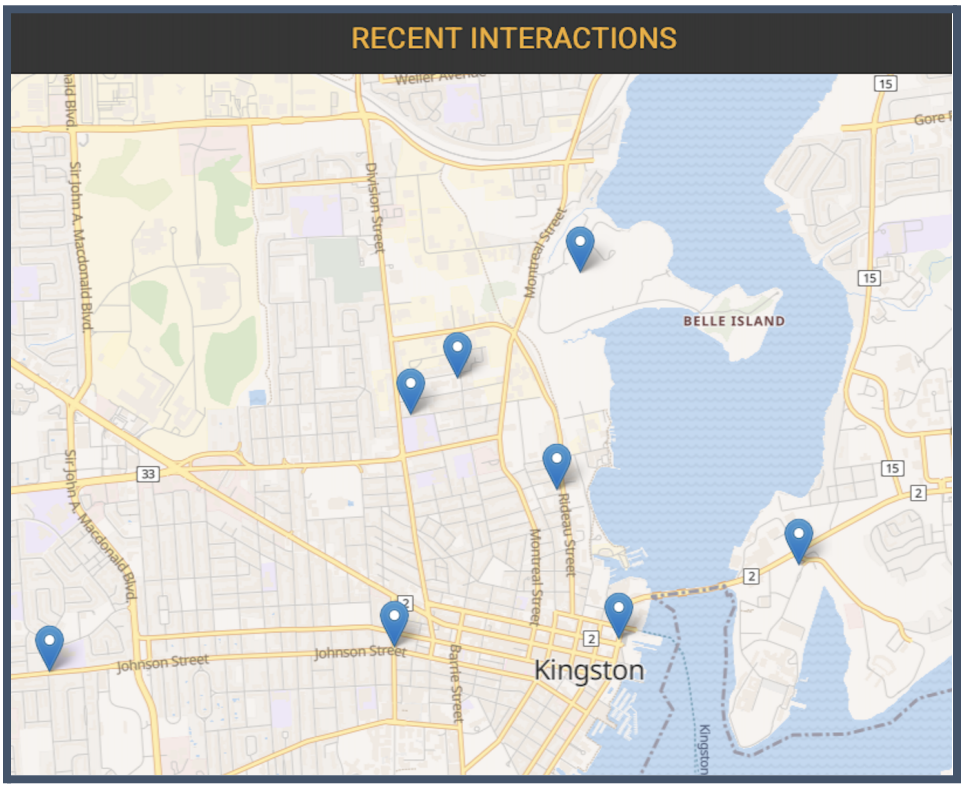

Figure 1. Screenshot of the map on the homepage, with multiple pins displayed.

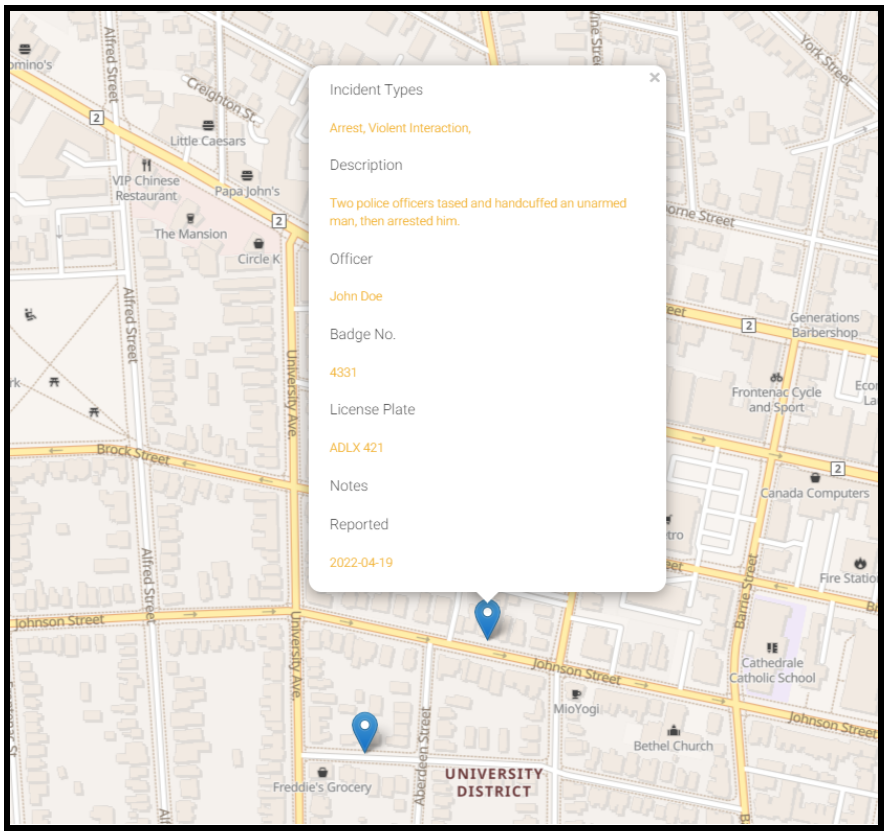

Figure 2. Screenshot of pin details popup, shown after clicking on any pin.

A user may report a new incident by clicking anywhere on the map. They are then redirected to a page with multiple input forms to fill out with the aforementioned details of the incident. Many of these forms are optional in case a user does not know all the relevant information about their incident, or if they do not wish to disclose them. Once submitted the app saves the new incident report to the database, creating a new pin on the map at the location originally chosen by the user. That pin becomes immediately visible to all users on the site.

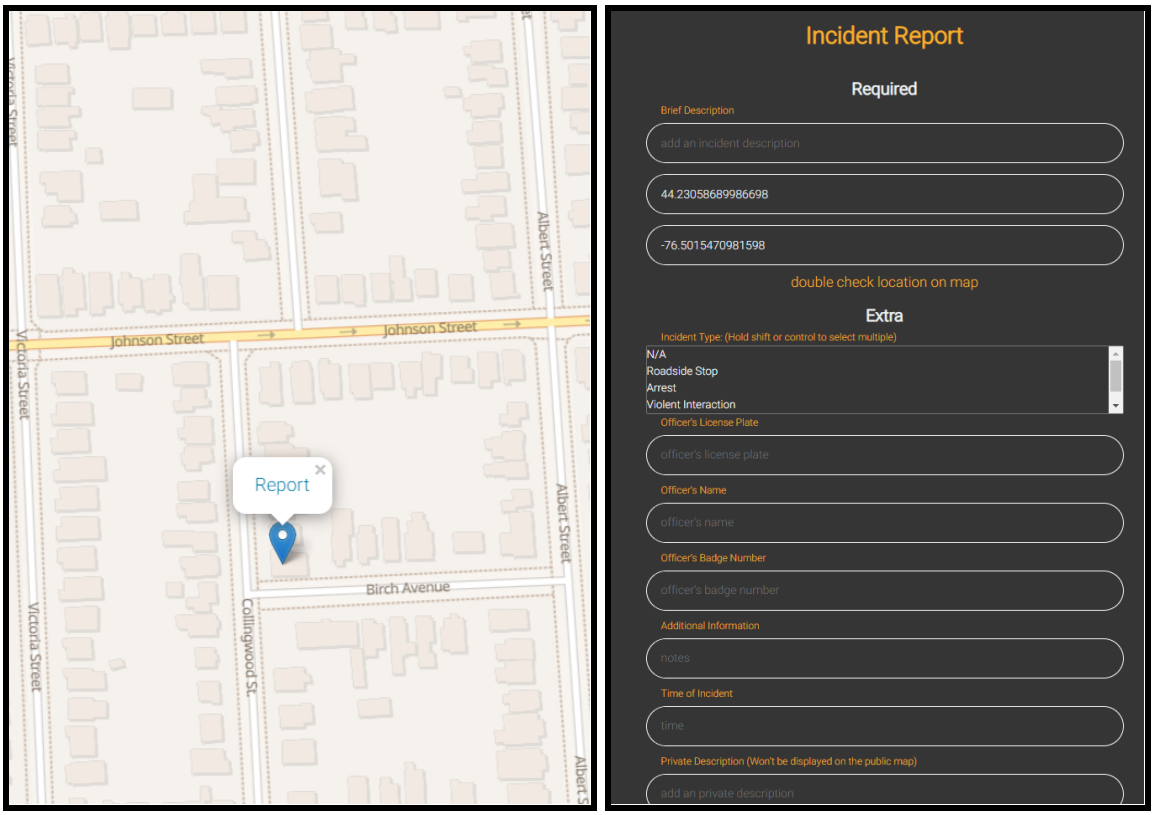

Figure 3. 'Report' link that appears when clicking on the map, and the incident reporting page that it links to.

Incident reports may also be viewed in a text based list on the 'Incidents' page, should the user not want to view the full map. The 'My History' page further allows the user to see a list containing only the incident reports that they uploaded. Currently this is the only method by which users may search and/or sort the incidents list. A more comprehensive search/sort functionality will need to be implemented in the future by whoever inherits this project.

The app also has a functional user account registration and login system. The app does not record any identifying information about user accounts besides the email used. Note that emails are not verified with unique tokens in the current version of the app. This will need to be implemented along with a password reset system by a future team.

An account is not strictly required in order to upload incident reports. Thus users may upload reports fully anonymously if they so choose. The database internally distinguishes between reports uploaded from registered accounts and those from guest users. This is accomplished through the user_id value that is attached to every incident report. Reports from registered users will have a user_id corresponding to their account number (a positive integer), while reports from guest users will have a user_id of -1. This distinction will be important to the project's future as part of a report verification and filtering system. However the site does not display this distinction publicly on map pins at this time.

5.2. Core Libraries

The site is primarily implemented in Python 3 using the Flask library [1]. Flask itself is dependent on Jinja [2] for rendering dynamic HTML templates, and Werkzeug [3] for its core web server gateway interface utilities. Both the front and back ends of the site are written in Python 3. Flask was chosen for its simplicity and accessibility as a framework for building web applications. Additionally, Python is a relatively beginner friendly language. Since the plan for this project was always to pass it on to the community, who may not have extensive programming knowledge, we felt it was important to choose a language and framework that would be easy for beginners to work with.

The database is implemented using the SQLite [4] engine. SQLite was chosen because it is lightweight, fast and easy to set up. Once again it was important that we chose a database library that would be relatively simple for novice programmers to learn, and SQLite fit the bill. The Flask-SQLAlchemy [5] library provided the features necessary for the Python backend to interface with the database. This library was also used to define all the core data models used in the database.

The site's flagship feature, its interactive map, was implemented in JavaScript using the LeaftletJS [6] library. Leaflet was chosen for its mobile friendly interface, comprehensive global street map data, and because it integrated with the rest of the project easily. Furthermore, the map pin features it provides are precisely what the original requirements requested. Although JavaScript is unfortunately not the most beginner friendly language, it was by far the most straightforward method of implementing an external mapping library. LeafletJS also has very extensive and well written documentation, which should make it easier for our successors to learn its features.

Additionally, the project uses the SeleniumBase [7] API to handle automated testing through the pytest [8] library. While the project contains all the necessary frameworks for running any number of automated tests, we did not create many test cases for it to run. Resultantly the project's current automated testing features are heavily limited at this time. This functionality primarily exists so that it can be easily expanded upon, should automated testing become necessary in the future.

5.3. Organization

We used a Github repository to host the project's source code and to manage code collaboration [0]. Our source code was initially forked from a Python Flask application that we made in 2021 for another course: CISC327 - Software Quality Assurance. It has since been heavily modified to fit this project's distinct requirements. Resultantly our project consists almost entirely of original code written this year, though it still follows the organizational structure of the original template. The template was provided by the course's instructor, Dr. Steven Ding, who

also wrote extensive documentation about the app's file structure and data flows, as well as instructions for how to run the application. This documentation can be found in the README files on our Github page, edited to compensate for this project's differences.

5.4. Page Design : HTML & CSS

iWitness was designed with the interactive map as a priority. Visualizing local interactions or documenting new ones would be the average user's first preference when opening the application. Therefore, we thought having the map on the landing page as well as incorporating the reporting functionality into the map was the most intuitive and efficient option.

Since our community partners requested more than just a landing page it was imperative users could seamlessly move between the different pages. Options such as a drop down menu were considered, however, after testing a couple options we settled on a navigation bar at the top of the webpage. The navigation bar allowed all the options to be continuously displayed while blending in and providing maximum space for the map below.

Finally, to encourage adoption it was critical iWitness looked modern and professional. To that end many different designs and colour schemes were tried with varying success. Initially, to pay homage to the community group we tried a colour palette consisting of purple, yellow, green, and red. This UI (Figure 4) was presented at our midterm review, unfortunately, this version looked both amatuer and uncoordinated. Our second iteration (Figure 5) had a blue, yellow, and red colour scheme. These colour worked well together, however, came across as too police centric and potentially unsettling for individuals with bad experiences with police. The chosen design (Figure 6) was inspired by *Watchmen* and uses black and yellow to emulate a searchlight. Since the users are using this application to keep *watch* over their communities we thought the design was more appropriate while also looking sleek and contrasting well.

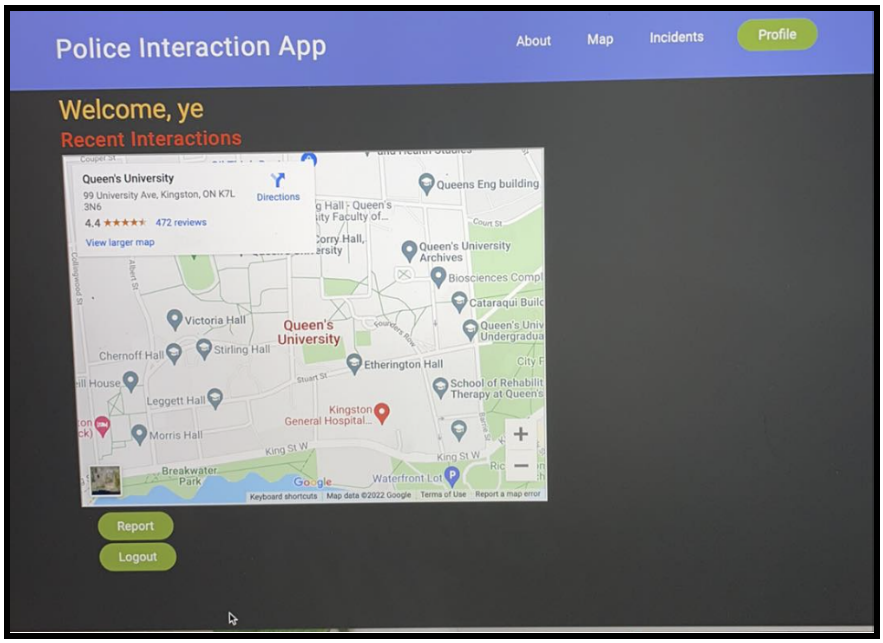

Figure 4. Screenshot of the landing page with our first user interface design.

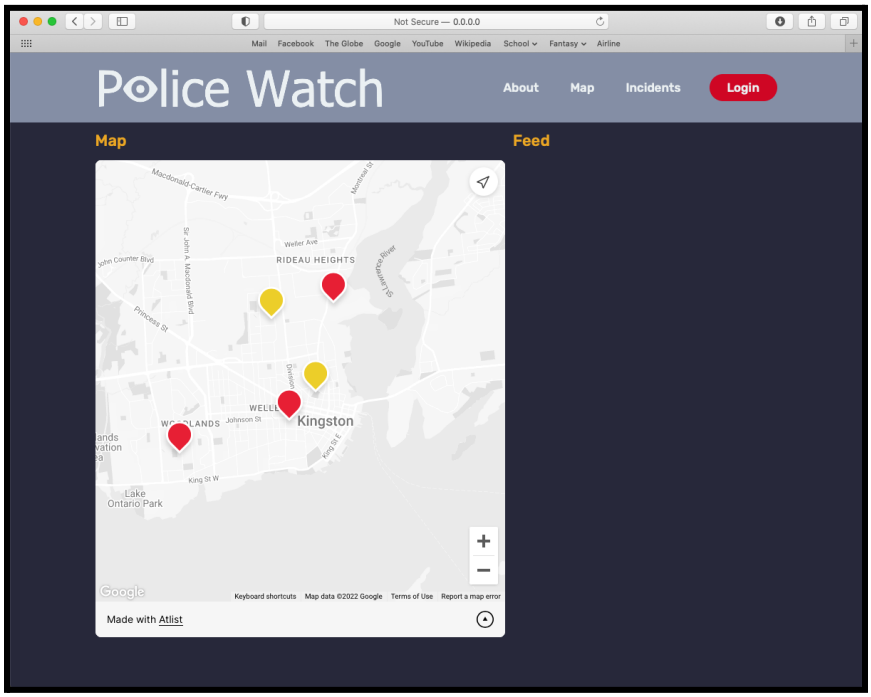

Figure 5. Screenshot of the landing page with our second user interface design.

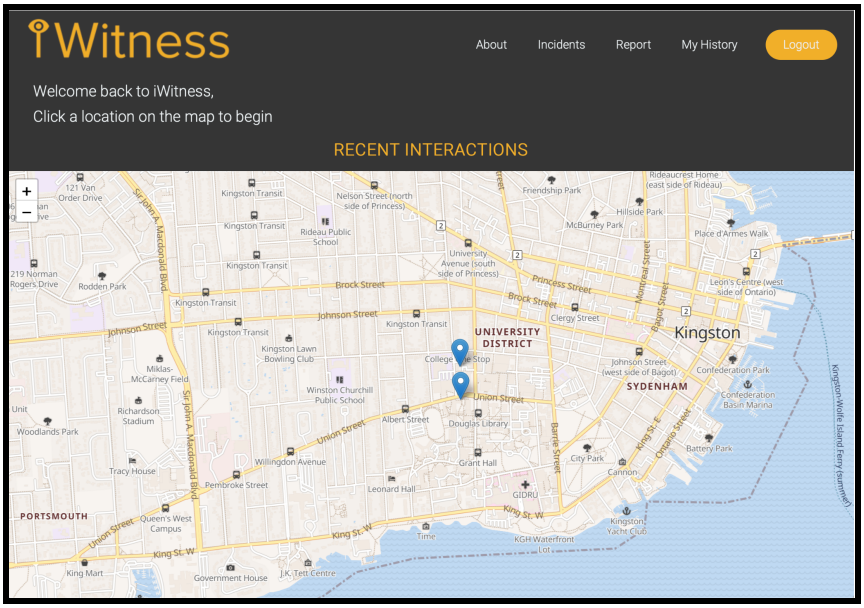

Figure 6. Screenshot of the landing page with our final user interface design.

6. Discussion

6.1. Results

Over the semester, we were able to develop a web application that is able to catalog, and map reported incidents with police. Although the app is intended to be accessed from a browser, there is currently no publicly accessible website to host the app, as these hosting services generally charge a fee. However, thanks to the Flask library, setting the app up once a site is available should be very easy.

Do we think our app is ready for use? Well, it depends on the usage of it. For a small community, we believe the app is a very effective tool. With a team of people monitoring the app, we think they can use the app to its full capabilities, while watching to ensure it is being used correctly. However, we do not think the app is ready to be used publicly around the world. Firstly, its performance has not been extensively tested with large data sets. The app works perfectly if focused on one neighborhood, but if it was storing and displaying thousands of reports from multiple cities, it may begin to have issues. Another issue to consider with public use is spam and false information. As it stands, these issues are easy to solve on a small scale, as false reports can be removed manually. However, on a large scale it would be hard to manually monitor every single report that is entered. We would be comfortable with releasing this app to the community that requested it for their own personal use of it, but it would require more work before releasing it worldwide as a public tool.

6.2. Potential Features & Improvements

During the development of the app, we prioritized the features that we thought were most important to the functionality of the app. However, due to the limited time frame for the project, we weren't able to fully implement a lot of the features that we wanted to.

For example, we would like to create a more mobile-friendly interface for the app. Currently, the app is built with desktop users in mind, however the app could benefit greatly from mobile use. Because of the app's use of location data, it would be very convenient to be able to record incidents on site by simply pulling out a phone and filling out the data. Although this is possible by accessing the app through a mobile web browser, the app's interface isn't built with mobile use in mind. To accommodate this, we would like to implement a separate design for mobile screens, and also implement mobile-friendly controls such as swipes and multi-touch.

Another feature we would like to implement is some type of verification or moderation feature. Currently, the app is accessible to everyone, with no account needed in order to report an incident. Although accessibility is important, there is a risk of people using the app maliciously by creating fake incidents, or entering false information. The app does distinguish between reports made by a guest and those made by a registered user, but outside of that, we have no way of knowing whether a report is genuine or not. This issue is difficult to fully solve, and still affects a lot of mainstream social media sites today. One potential solution to this is manual moderation. Since the goal is to hand the app over to the community that requested it, we can implement tools that allow them to have control over the app. Since the app is intended for use on a small scale, they can create a team to monitor the app, and delete any reports that they suspect are fake. Another solution is to implement filters that users can toggle to only show reports from registered accounts. This way, if they are being targeted by anonymous users, they can simply turn those reports off, and if they are still having issues, they can more easily identify which accounts are causing them.

Lastly, another feature that we would like to implement is extra security and encryption. We believe that privacy is very important, especially when sharing sensitive information about police interactions. Currently, recorded incidents on the app are publicly viewable. We would like to add a feature that allows users to determine who can view their reports, in order to better protect their privacy. Another important topic to consider is data encryption. Currently in order to register an account on the app, the user must provide a username and email. Depending on the username and email address used, this information could potentially identify users, which could create problems if data is leaked. Although we believe security is a high priority, we decided it was outside of the scope of the project. Since Tella was an initial consideration for use as a foundation for the app, it may be possible to have more work done to see if Tella's security can be incorporated into the app. Otherwise, additional research should be considered to find out if there are any data encryption tools that are simple to use and implement.

6.3. Limitations

One of the biggest limitations we experienced early on in development was selecting a platform for the app. Although the app was initially envisioned as a mobile app for Apple and Android devices, developing for these platforms has a few major drawbacks. One, they are less accessible. Both Apple and Android require professional licenses in order to create and release apps on their platforms. They are also more complex systems to create. If the end goal is to hand over the app to the community, they should be able to modify and update it without extensive coding expertise, and without purchasing expensive licenses. Another issue with developing for Apple and Android is that the two systems are not compatible, meaning two separate apps would need to be made for each platform. This was not likely to be possible to create within the 4-month timeline, and one platform would have to be sacrificed to focus development on the other. This is a major drawback since removing support for one of the platforms would make the app inaccessible for a large number of users. Although a web app has its own limitations, such as a reliance on internet connection, we thought it was the best option to make the app both easy to use, and easy to modify.

7. Teamwork

Over the course of the project, our team used Microsoft Teams for communication, and collaborated on the app through Github. Our team met every Tuesday at 3pm to discuss our plans and goals for the weak, and to discuss any issues we were encountering. We were also paired with a PhD student in philosophy, Sofie Vlaad, who works with Dr. Catherine Stinson. Sofie was available to discuss any general questions we had about the project, as well as facilitate our communication with Dr. Stinson to ensure the project's goals were staying on track. In exchange, we provided Sofie with some exposure to the world of computer science, since it was relevant to her interests and research. Over the course of the project, we had a few meetings with Sofie to discuss coding, such as what languages we will be using, as well as giving her a quick tour of the code. We also discussed a few simple tasks that would be possible for Sofie to contribute to such as implementing a character limit on report descriptions or checking that the user's form entries are in an acceptable format.

After our mid-term checkup, Dr. Stinson suggested that we divide our team into different roles, to make working more efficient. Aidan Turnbull was in charge of developing the app's front-end, which includes the HTML and CSS components of the web-app. Joshua Chai-Tang was responsible for the app's back-end, which includes designing the report's data structure, and making sure the data could be stored and retrieved easily. Anthony Rios was in charge of the mapping component of the app, which includes researching open-source mapping libraries, and implementing them into the app. Anthony was also in charge of communication between the team and the supervisor, providing a summary of the team's work for the weekly check-ins.

8. Experiences & Lessons Learned

8.1 Anthony

This project was my first experience working on something more large-scale instead of a small assignment for a class. Most people have personal projects that they work on, or have experience from internships, but I didn't have any of that experience yet. I had also never worked in HTML or CSS before. Over the term I became more familiar with it and eventually engaged with it more as I was integrating the map into the home page. I also realized throughout the term that my strengths were communication and organization. During our team meetings I would ensure we had a plan for the following week, keep us updated on upcoming deadlines, and check if there were any problems we needed to sort out. Overall, having to coordinate with a team to tackle a larger project helped me develop my teamwork skills.

8.2 Josh

For me this project was primarily a lesson in teamwork, coordination and communication. I found the greatest challenges to be in the early stages of the project. Figuring out how to approach the problem took a lot of research not only into the tools and libraries available, but also into the individual skills and resources of my teammates. Finding the right way to divide the work between ourselves took some time to get right as well. I've used Github for group projects before, but coordinating our code on this project proved more complex than I'd anticipated because of the greater level of dependencies and overlapping sections.

Communicating with our supervisor to balance the project's expectations with our actual abilities during the first few meetings also proved difficult. I was afraid of promising more than we were capable of, but still wanted to make sure we would deliver a useful product by the end of the term. This is the first time I've worked on a project where I've had to know my teammates this well in order to succeed, so it's been a major lesson in teamwork for me.

What I consider to be the most important lesson was learning how to take advantage of the support and resources available to us through the Ethics and Technology team. At first we approached this project with a very independent mindset, assuming we'd have to do everything ourselves like we've had to in other courses. It genuinely hadn't occurred to us that we could ask for help and feedback from the larger group. As soon as we learned to start asking more questions during lab meetings, progress on the project sped up considerably. Guidance from Sophie and suggestions from Rohan in particular turned out to be incredibly valuable for us. I'm confident that this lesson will be the most useful to me in my future career, where I'll likely be surrounded by co-workers and company resources that can help me.

8.3 Aidan

This application was the largest computing project I have contributed towards during my undergrad at Queen's. Both my overall understanding of how web-applications are built and HTML/CSS skills have improved significantly. Before this project I knew how to code in Python, HTML, and SQL independently, however, this was my first experience putting it all together and seeing how the different components interact with each other. Despite my technical growth by far my biggest takeaway from this experience was the development of my *soft* skills and ability to work effectively with a supervisor.

Towards the beginning I took our miscommunications with the direction of the project personally. Upon reflection, I was too fixated on our original vision for the project that it prevented me from adapting, taking direction, and ultimately making progress on the application. Although there was never animosity amongst the group, looking back my approach was indirectly handicapping our ability to get to the coding stage. After receiving a humbling email from our supervisor it clicked that I had to change course. Through gritted teeth I decided to talk less and let our work speak for itself. Although we still had our ups and downs following that moment, the project made a lot more progress afterwards. I made a conscious effort to follow directions, not make excuses, and let others speak first. Looking back, it is these soft skill lessons that I will remember most and carry with me into the professional world.

9. Conclusion

Police harassment against targeted racial and ethnic groups is a serious problem affecting many cities across Canada. With the discussion about police funding and abuse of power popping up all over mainstream media, many communities are looking into their options and how they plan to tackle these issues locally. At the request of a community group, our team built a web-based app that is able to catalog and map interactions with police officers. With the help of this app the community can gather evidence, and look at trends to determine if their local police are targeting their neighborhood. The community can then look at their collected data to determine their next steps against this issue. With more work and updates, we believe the app has potential to be released to more communities and organizations who are interested in taking a stand against abuse of power in their neighborhoods.

10. References

Coding Documentation

[0] iWitness Github Page : https://github.com/chai-tang/Police-Interactions-App
[1] Python Flask Documentation : https://flask.palletsprojects.com/en/2.1.x/
[2] Jinja Engine Documentation : https://jinja.palletsprojects.com/en/3.1.x/
[3] Werkzeug Utilities Documentation : https://werkzeug.palletsprojects.com/en/2.1.x/
[4] SQLite Documentation : https://www.sqlite.org/index.html
[5] Flask-SQLAlchemy Documentation : https://flask-sqlalchemy.palletsprojects.com/en/2.x/
[6] LeaftletJS Documentation : https://leafletjs.com/SlavaUkraini/
[7] SeleniumBase Documentation : https://seleniumbase.io/
[8] pytest Documentation : https://docs.pytest.org/en/7.1.x/

Research Bibliography

[1A] The concrete effects of body cameras on police accountability : https://theconversation.com/the-concrete-effects-of-body-cameras-on-police-accountability-171460

[2A] Why Police Departments Don't Always Release Body Cam Footage : https://time.com/4453310/milwaukee-police-sylville-smith-body-cams/

[3A] New OHRC report confirms Black people disproportionately arrested, charged, subjected to use of force by Toronto police : https://www.ohrc.on.ca/en/news_centre/new-ohrc-report-confirms-black-people-disproportionately-arrested-charged-subjected-use-force

[4A] Perceptions of and experiences with police and the justice system among the Black and Indigenous populations in Canada : https://www150.statcan.gc.ca/n1/pub/85-002-x/2022001/article/00003-eng.htm