Curtain

by

Kai Dragomir, November Harris, Charlie Sims, and Lily Watterson

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology/Cybersecurity

Kai O. Dragomir	4/15/2022
Kai Dragomir	Date
November Harris	4/15/2022
November Harris	Date
Charlie Sims	4/15/2022
Charlie Sims	Date
Lily Watterson	4/15/2022
Lily Watterson	Date
Tyler Hopperton	4/15/2022
Tyler Hopperton, Faculty Advisor	Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2022

Table of Contents

List of Illustrations

Abstract

According to Peacock[1], data collection methods are becoming increasingly invasive. The preservation of information is shifting from a technological problem to a moral dilemma. As a result of this growing problem and concern, we decided to work on a solution addressing this. Our solution's aim would give back some control to users who wish to still use the internet, without having as much of their data collected. By researching and developing different tools that protect a user's data and privacy, we were able to develop Curtain. Our application, Curtain, serves to provide a centralized platform for privacy tools. There are several tools that we implemented into our application: blocking web tracking (including data such as user behavior in and outside of a website), blocking ads, bypassing AdBlock detection messages, deleting cookies, and a dashboard for easy access to settings and usage data. Our results and progress on the matter means that it is possible to centralize tools which made it easier for users to have more control over their privacy and data without needing to do additional research.

[1] Sylvia Peacock, "How Web Tracking Changes User Agency In The Age Of Big Data: The Used User", Big Data & Society 1, no. 2 (2014), doi:10.1177/2053951714564228.

Introduction

Project Summary:

A suite of tools that will work together to ensure one's data privacy.

Problem Statement:

According to Kalavri et al[2]., web trackers monitor user behavior on the web and this information is used for customized and targeted advertisement. Such behaviors include the following: page fetching —when a page is fetched by a browser, which makes an HTTP request to the site for a URL; execution context — refers to the environment in which the code is run in, in the websites which includes user input such as mouse clicks, hovering, and typing; and client-side storage — storing the state of a website (typically cookies). While tracking can provide numerous benefits such as personalization, targeted advertisement, and site analytics. The usage of tracking, however, unsurprisingly comes at the cost of privacy[3]. Roesner et al.[4], investigated the current counter measures against blocking tracking by disabling JavaScript, third-party cookie blocking, pop up blocking, and browsing the web by starting in incognito mode. It is also worth noting that there are two forms of tracking, within-site and cross-site.

Companies are beginning to use social media data and web tracking technologies that store personal data on an unprecedented scale[5]. The scale of the usage of this data has introduced challenges to the internet user(s), primarily concerns about privacy and data. Most online corporations operating in the online market do not self-regulate because it puts them at an economic disadvantage. Information property rights researchers are advocating for micro payments to reimburse people for the use of their personal data, examples[678] of this practice are seen. However, this approach fails to understand that the issue of preservation of information is shifting from a technological problem into a moral dilemma[9]. The current online exchange is not between two informed agents, nor does it improve the online public sphere, and this exchange is not equivalent. According to Peacock[10], this falls under an unconscionable contract — exploitative, unjust, unavoidable, and puts the burden of an economic transaction wholly on one side, and in this case the online user. For most users, the inherent unjust and unequal burden is nowhere spelled out, and even if it were, people would be unlikely to read the terms.

---

[2] Vasiliki Kalavri et al., "Like A Pack Of Wolves: Community Structure Of Web Trackers", PAM 2016: Passive And Active Measurement 9631 (2022): 42-54, doi:https://doi.org/10.1007/978-3-319-30505-9_4.

[3] Franziska Roesner, Tadayoshi Kohno and David Wetherall, "Detecting And Defending Against Third-Party Tracking On The Web", 9Th USENIX Symposium On Networked Systems Design And Implementation 12: 155-168.

[4] Roesner et al. 158-160

[5] Sylvia Peacock, "How Web Tracking Changes User Agency In The Age Of Big Data: The Used User", Big Data & Society 1, no. 2 (2014), doi:10.1177/2053951714564228.

[6] Eirik Furubotn and Svetozar Pejovich, "Property Rights And Economic Theory: A Survey Of Recent Literature", Journal Of Economic Literature 10, no. 4 (1972): 1137-1162, https://www.jstor.org/stable/2721541.

[7] Jaron Lanier, Who Owns The Future? New York: Simon & Schuster, 2013.

[8] Jessica Leber, "A Dollar For Your Data", MIT Technology Review, Last modified 2022, https://www.technologyreview.com/2012/06/08/185769/a-dollar-for-your-data/.

[9] Erez Aiden and Jean-Baptiste Michel, Uncharted, 1st ed. New York: Riverhead Books, 2013.

[10] Sylvia Peacock, 22-25

Solution:

Our solution is to create a suite of tools which will be integrated in a centralized plugin to automate things that an end user(s) wants to do with their data. The tools would funnel up statistical data to a visibility dashboard to ensure that the suite of tools work. The dashboard would include elements blocked, what is being blocked, cookies declined, additional settings for the tools, etc. Elements which will be blocked will include pop-ups, some background scripts, third-party advertisements, etc.

Project Source:

November originally conceived the idea; they went onto a website which prompted to ask about cookie settings to be declined or accepted. November wished that there was some sort of automated tool to decline cookies. This idea was suggested in a group meeting, where we unanimously agreed that this was a pressing concern and issue regarding one's personal data and identity online.

The conversation then turned into how we could implement a solution to fix this concern as well as address other pressing concerns in privacy and data management online. Our proposal based on some information we had at the initial conception was creating some form of a centralized plugin that would make it easy for users who are not technically aware of what is happening in an online sphere to control more of their data that was being collected on them.

The team originally formed by talking about ideas prior to this course which made us realize we have similar goals. We had come up with an idea for some application to inform users of where public bathrooms are, a mental health resource, and an application that would serve to be a form of safeguarding an individual upon being inebriated (i.e., preventing drunk texts and calls). So, we decided to form a team together, upon seeing that we had similar goals in terms of what we wanted to accomplish from the course.

Discussion

Project Objectives/Goals:

This section will highlight some of the major objectives and goals that Curtain aims to accomplish. These goals will make it possible to accomplish giving back the user's control in the online sphere where there is a prevalent concern of not being able to consent.

- Increase the end user's data privacy control
- Increase accessibility, making it to where there is a centralized tool
- Display what has occurred to be transparent

Project Scope:

"Our team will develop a centralized suite of tools that enables users to have more control over their data privacy."

Quick Project Timeline:

The project timeline (*Table 1: Project Timeline*) here will break down the scope of our project into measurable tasks. Curtain will follow the table to ensure that objectives and goals of the project can be met in a timely manner.

Table 1: Project Timeline

Task #	Task Name	Duration	Start Date	End Date
1	Conception	150 hours	8/2/2021	8/30/2021
2	Market Research	60 hours	8/30/2021	9/13/2021
3	Research	200 hours	9/13/2021	4/11/2022
4	AdBlock Implementation	20 hours	9/14/2021	10/1/2021
5	Advisor Meetups	16 hours	9/13/2021	4/23/2022
6	Testing for presentation	50 hours	10/18/2021	11/10/2021
7	Prepare for Presentation	10 hours	12/4/2021	12/11/2021
8	Tracker removal implementation	90 hours	12/12/2021	1/10/2022
9	Auto-Cookie Delete Implementation	20 hours	2/12/2022	2/21/2022
10	Other desired add-ins and bugs	200 hours	1/10/2022	3/1/2022
11	Bug fixes	200 hours	3/2/2022	4/3/2022
12	Prepare for IT Expo	10 hours	4/4/2022	4/11/2022

Technologies Used:

This section will detail the technology used to communicate and develop for the project. When considering what technology to use, it is important to consider what one knows and what one would want to learn. In our case, we were familiar with a few of these technologies and had to learn a new language. This section will also include technology we used to collaborate with one another, which is not necessary for the development but contributed to the developmental process.

Main Developmental Technology:

- JavaScript – this was used for the logic behind the application
- Chromium – this engine browser which was reversed engineered to figure out how to fit the said tools
- HTML – this was used for creating an interface to allow the user to interact without needing additional information
- Bootstrap – this was used to design further design the interface for aesthetic purposes
- UNIX File server – used to hold files which were used by the plugin to apply mass settings

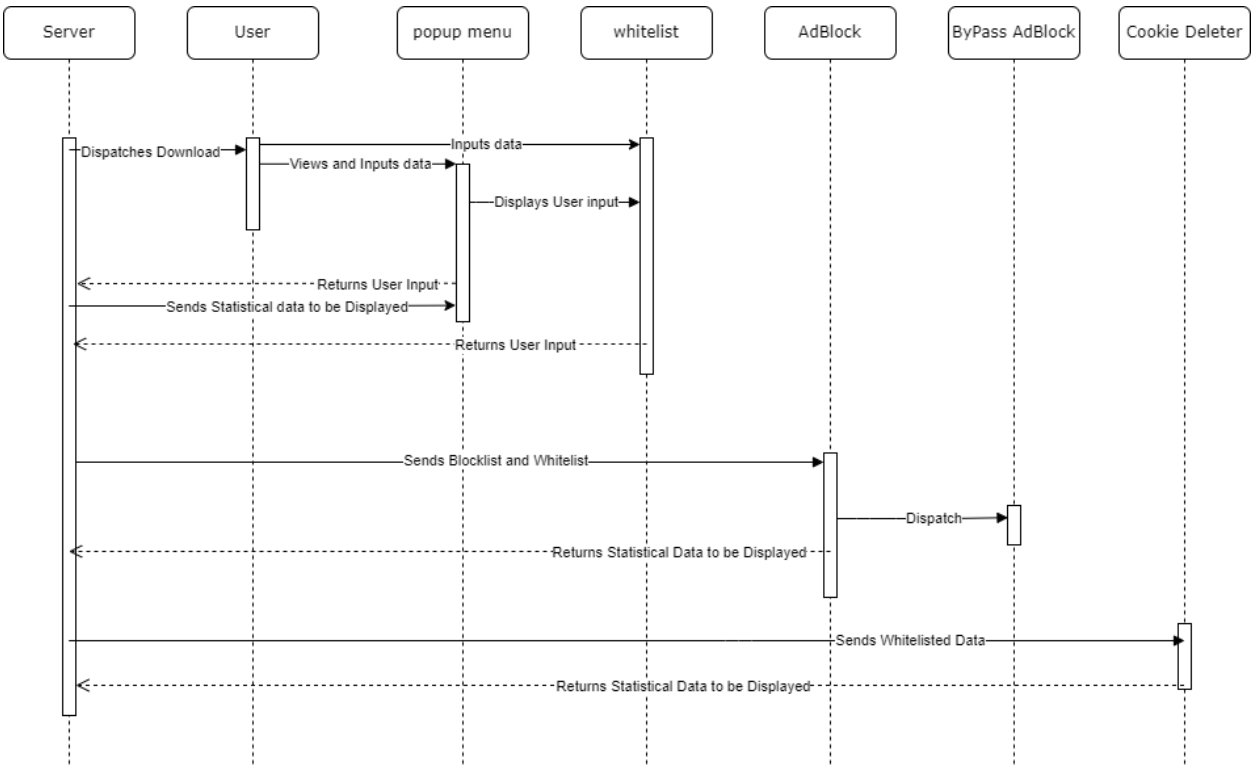
Collaborative Technology not Vital:

- Github – this was used to collaboratively work on code.
- Trello – this application was used to help collaborate with other people on the team and give an insight on where individuals were
- Discord – this application was used to primarily communicate with each other, the usage of channels and pinned messages made it easy to communicate, share information, and refer to information.

Technical Architecture Diagram:

We put a lot of thought into our design and layout of our application to account for a good user experience while using our application. A good design will not only attract users to our application, but also make it to where we have loyal users who come back to our application. In *Figure 1: Sequence Diagram*, will show a sequence diagram for our application at the current state which Curtain's current development is at. This diagram will show interactions that occur between the different components during the usage of our application.

Figure 1: Sequence Diagram



User Personas:

In the tables below, *Table 2: User Persona: Paranoid Perry* and *Table 3: User Persona: Regular Ruby*, illustrate the two personas which will show two different types of behaviors we are expecting to have interact with our application.

Table 2: User Persona: Paranoid Perry

User Persona: Paranoid Perry	
	Title: Security Analyst
	Name: Perry Jones
	Age: 25-34
	Gender: Man
Behavior:	Attempts to reduce footprint online by using multiple means.
Pain:	When technology does not work as expected
Needs & Goals:	Securing the workspace and their own personal computer

Table 3: User Persona: Regular Ruby

User Persona: Regular Ruby	
	Title: Sales Manager
	Name: Ruby O'Malley
	Age: 25-34
	Gender: Woman
Behavior:	Accepts ToS, notification, scripts, and cookie requests upon initial request upon visiting a website
Pain:	When technology does not work as expected
Needs & Goals:	Securing the workspace and their own personal computer

Use Cases:

In the tables, below, *Table 4: Use Case ID 1* and *Table 5: Use Case ID 2* showcase the two minimum features of expected user interaction of our application. The tables represent the minimum features of user interaction required to function as initially planned. We added another feature for the convenience of the user, but it is not a required function or interaction to utilize Curtain. This feature is explained in *Table 6: Use Case ID 3*.

Table 4: Use Case ID 1

Use Case ID	1
Use Case Name	Change Settings
End Objective	Changes the Settings of what is happening
User/Actor	End user
Trigger	Confirming changes in the setting menu of the plugin
Frequency of Use	Occasional
Preconditions	Setting page is operation as other parts of the plugins are as well
Basic Flow	1. User interacts with the dashboard to show the menu/option(s) available 2. User clicks the setting icon 3. User modifies something in the settings, perhaps how something is displayed 4. User clicks a confirmation button 5. Settings are saved, tool(s), and the dashboard update accordingly.
Alternate Flow	1. User interacts with the dashboard to show the menu/option(s) available 2. User modifies settings on the dashboard 3. User clicks confirmation button 4. Settings are saved, tool(s), and the dashboard update accordingly
Postconditions	Settings are successfully applied, either in the dashboard appearance or a tool(s) usage.

Table 5: Use Case ID 2

Use Case ID	2
Use Case Name	Display statistics
End Objective	Displays the statistics from the tools that are enabled
User/Actor	End user
Trigger	When the user looks at the dashboard
Frequency of Use	Occasional
Preconditions	When the user clicks to show the dashboard
Basic Flow	1. User interacts with the dashboard 2. Dashboard displays specified statistics which have been set by the settings
Alternate Flow	1. User interacts with the dashboard 2. Tools communicate to server which has settings to display data 3. Data is displayed
Postconditions	Dashboard displays the data (assuming that the tools are enabled) of how many element(s) have been blocked.

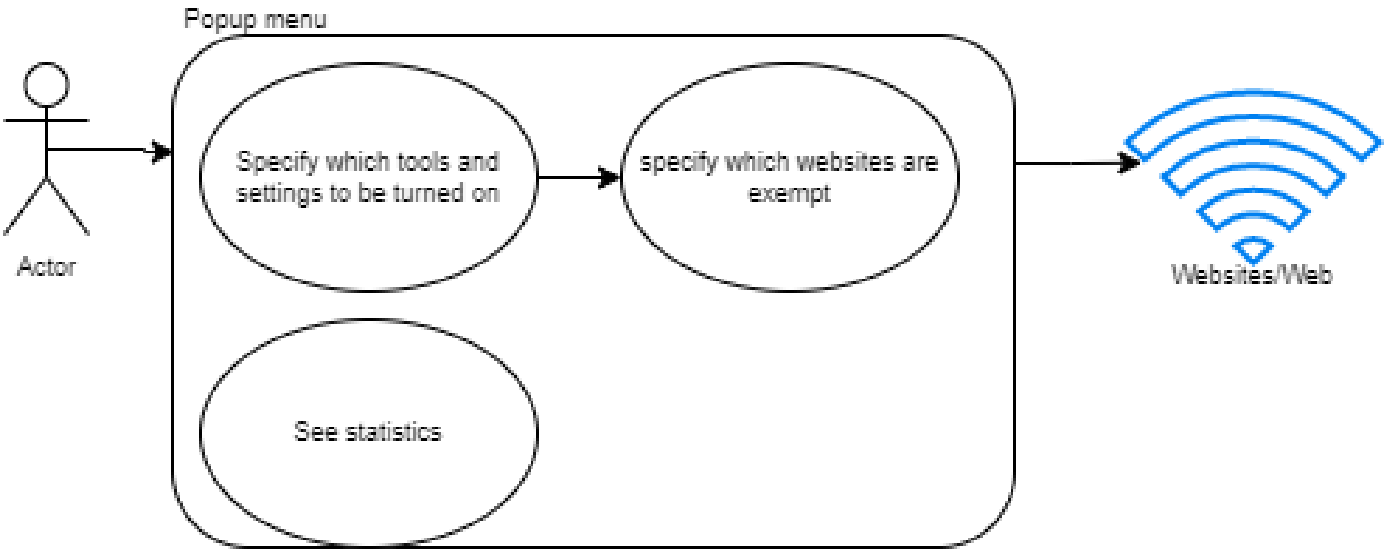
Table 6: Use Case ID 3

Use Case ID	3
Use Case Name	Update Whitelist
End Objective	Updates the Whitelist as intended
User/Actor	End user
Trigger	When the user adds an entry and saves the whitelist
Frequency of Use	Occasional
Preconditions	User must type in a URL and click the save button
Basic Flow	1. User interacts with the dashboard 2. User clicks on configure whitelist 3. User types in a URL 4. User clicks save
Alternate Flow	1. User interacts with the dashboard 2. Tools communicate to the server to verify specific URLs are not utilizing the said tools
Postconditions	Settings are successfully applied, either in the dashboard appearance or a tool(s) usage.

Use Case Diagram:

The diagram below, *Figure 2: Use Case Diagram*, displays the planned use case for Curtain's application. This diagram is expected to interact with the application, it will show how the application will be utilized and communicate internally to itself. The diagram represents the minimum features required to function as initially planned.

Figure 2: Use Case Diagram



Testing Plan:

Overview

This section contains information on how the Curtain plugin was tested and evaluated at the current state of the development of Curtain. The methodology which was chosen for this application was initially agile; however, overtime the methodology became a mix of waterfall and agile. This change was because of life conflicts and plans not being able to be followed as planned initially and needing to change the development lifecycle while still attempting to stay on track.

Methodology

The approach was to have the developers initially test their new feature or component, in their development space. Then have other people on the team test it in their spaces as this would provide different environments, i.e., testing it on different browsers, how different users would act on the internet, and local variables. Then after the manual testing was done, we were hoping to develop an automated testing instance of the test. To ensure that previous developments were still compatible with our current developments.

However, this approach was seen to be too slow of an approach to test adequately. We then switched to having the developers test their new feature or component then having a singular person who tested the developer's branch on different browsers. In addition to having numerous unpredicted events which greatly stunted our development process. We also realized that writing automated test cases were not possible with some tests, e.g., AdBlock. As there was no clear way to assure that ads were not showing.

We also found that developers who tested their own code were unlikely to find issues compared to someone who did not develop the code. So, having another individual test someone's features or branch assuming that the branch itself had already undergone some form of developmental tests, e.g., to see if the code would compile or run without errors. This allowed us to see and catch bugs sooner to work on fixing the said bugs found.

Scope

1. Unit Testing
    a. AdBlock
    b. Bypass AdBlock Detection
    c. Popup menu
    d. Dashboard
    e. Cookie delete
2. Integration testing
    a. AdBlock/Popup menu
    b. AdBlock/Dashboard
    c. Cookie delete/Popup menu
    d. Cookie delete/Dashboard
3. Security Testing
    a. Analysis of what can be broken into
4. User Acceptance Tests

Unit tests (see *Table 5: Unit testing: AdBlock, Table 6: Unit Testing: Bypass AdBlock Detection, and Table 7: Unit Testing: Popup Menu* for details*)* will be done to ensure that the individual tools can be used correctly and properly. Integration testing will be testing how the tools themselves work together with the other tools. Security testing (see *Table 8: Security Testing: Analysis* for more information) will be done to ensure that the plugin is secure. These features are tested because they are the minimum features which would align with what we initially planned for our tool to work. User acceptance tests (see *Table 9: User Acceptance Tests*) have been conducted by some end users to also ensure that other bugs which may have been overlooked are corrected and to ensure that the plugin itself is accessible.

Objectives

a. All major features and use cases need to be accounted for
b. All use cases must account for all the user roles
c. All bugs need to be found and attempted to be resolved beforehand

Test Logs and Procedures

Table 7: Unit Testing AdBlock

Test Case #	Test Description	Web Browser	Expected Result	Actual Result	Pass/Fail	Notes
1	Test AdBlock	Chrome	No ads seen	No ads seen	Pass	AdBlock bypass detection was active, which may have led to not seeing a window about the block
2	Test AdBlock	Edge	No ads seen	Advertisements were seen	Fail	Potential issues with how the detection are occurring, perhaps there needs to be a different detection/removal of ads
3	Test AdBlock	Brave	No ads seen	Advertisements were not seen	Pass	Potential issues with how the detection are occurring, perhaps there needs to be a different detection/removal of ads
4	Test AdBlock	FireFox	No ads seen	Plugin was initially showing up as corrupt. Additional steps were needed to be done.	Fail	There seems to be an overlapping issue of how the browser engine is run and structured to where it is not possible to pursue development at this time with it.
21	Test AdBlock	Edge	No ads seen	Advertisements were not seen	Pass	There seems to have been an issue with parsing the data with Edge, which has since then been fixed.

Table 8: Unit Testing Bypassing AdBlock Detection

Test Case #	Test Description	Web Browser	Expected Result	Actual Result	Pass/Fail	Notes
5	Test Bypass Detection AdBlock	Chrome	No popups asking to whitelist or turn off AdBlock seen	No popups asking to whitelist or turn off AdBlock seen. However, there was some delay in loading some of the pages	Pass	The delay will likely need to be fixed. Sometimes it causes a website to forcibly refresh.
6	Test Bypass Detection AdBlock	Edge	No popups asking to whitelist or turn off AdBlock seen	No popups seen asking for a whitelist or turn off AdBlock	Pass	The delay will likely need to be fixed. Sometimes it causes a website to forcibly refresh.
7	Test Bypass Detection AdBlock	Brave	No popups asking to whitelist or turn off AdBlock seen	No popups seen asking for a whitelist or turn off AdBlock	Pass	The delay will likely need to be fixed. Sometimes it causes a website to forcibly refresh.

Table 9: Unit Testing Popup Menu

Test Case #	Test Description	Web Browser	Expected Result	Actual Result	Pass/Fail	Notes
8	Testing Switches	Chrome	Specific tools turn on and off	The tools are not currently fully integrated	Fail	The tools are partially integrated, and it is difficult figuring out how to activate particular tools
9	Testing Switches	Edge	Specific tools turn on and off	The tools are not currently fully integrated	Fail	The tools are partially integrated, and it is difficult figuring out how to activate particular tools
10	Testing Switches	Brave	Specific tools turn on and off	The tools are not currently fully integrated	Fail	The tools are partially integrated, and it is difficult figuring out how to activate particular tools
15	Testing dashboard	Chrome	Dashboard shows up on activation from the popup menu	Dashboard shows up	Pass	The dashboard shows up, however is currently populated with skeleton code
16	Testing Dashboard	Edge	Dashboard shows up on activation from the popup menu	Dashboard shows up	Pass	The dashboard shows up, however is currently populated with skeleton code
17	Testing Dashboard	Brave	Dashboard shows up on activation from the popup menu	Dashboard shows up	Pass	The dashboard shows up, however is currently populated with skeleton code

Table 10:Unit Testing Cookie Delete

Test Case #	Test Description	Web Browser	Expected Result	Actual Result	Pass/Fail	Notes
18	Test Cookie Deleter	Chrome	Cookies delete automatically	Cookies delete automatically	Pass	Cookies deleted a little bit slower on this browser as opposed to the others
19	Test Cookie Deleter	Edge	Cookies delete automatically	Cookies delete automatically	Pass	Cookies deleted relatively quickly on this browser
20	Test Cookie Deleter	Brave	Cookies delete automatically	Cookies delete automatically	Pass	Cookies were deleted as expected

Table 11: Security Testing Analysis

Test Case #	Test Description	Weaknesses	Acceptable?	Notes
8	Analysis of Plugin vulnerability	The only thing which would be vulnerable is the server. Which did not fail any of penetration testing which was attempted on it.	Yes	Tested with xss, LDAP, DDOS, file injection, and command injection. PHP and SQL injections are not privy to the server.
22	Reanalysis of Plugin vulnerability with new tools added	Tested the whitelist component to see if it may have been privy to some database attack. It was not.	Yes	Retested the previous security test, with the new features and additions onto the plugin. It was found to not be privy, the whitelist logic had rightfully extracted and sent information to the server.

Table 12: User Acceptance Tests

Test Case #	Acceptance Requirement	Test Result		Notes
		Acceptable	Rejected	
11	Popup menu options are intuitive	15	0	Was commented on how simplistic the popup menu was
12	Popup is accessible	15	0	The simplistic nature of the popup menu made it accessible to other individuals
13	Plugin turns on and off tools as expected for	10	5	There were some issues with some browsers
14	AdBlocker and the warning to whitelist AdBlock is bypassed and functional	15	0	AdBlock seems to work without issues currently

Testing Review

We learned that it is rather difficult to implement a way to automate the tests, as the tests would not be able to verify that the AdBlock initially worked because sometimes the element is still on that page, despite not being visible. Not to mention, automated tests would not be able to return the errors which we were looking for. In the future, we believe that the product or tool needs to be fully finished before it can be automated and have some method of having a computer realize that a test had fully succeeded, i.e., being able to run through a sequence of commands to determine if something was visibly there was not an accurate way to determine if a tool had worked. We also learned the importance of replicating the tests on different environments.

Undergoing a phase of developmental testing made us realize that how different developing on different debug versions of the plugin was for browser engines, e.g., Gecko and Chromium. We especially had difficulty with Gecko's testing and development because of the difference in browser infrastructure being too large for us to continue to develop for both browser engines.

We also learned from receiving criticism from a design standpoint from other individuals. For example, feedback that was received was increasing the accessibility, i.e., some of the stuff that we had initially created for our popup menu was not the most intuitive. For example, the abandoned "gray-listing" feature would have made it possible for the user to temporarily revoke the usage of particular websites for the session without increasing their data vulnerability by turning the extension off on those sites. This was intuitive to the developer but for most individuals in the user acceptance test the gray listing feature was confusing and an unknown concept. Upon hearing this feedback, we asked if we had removed that feature would individuals have been able to navigate the rest of the plugin without issues. The answer was a unanimous yes, so this feature was removed.

Change Management Plan:

People who can request a change in the future, would be the users who use Curtain and security professionals. The goal of Curtain is to improve the control of a user's privacy and presence online. However, if the users want a particular feature or the removal of a feature, they are more than welcome to request it as they are the people who are directly impacted by using Curtain. Thus, they would have a direct impact on the matter. Security professionals, on the other hand, would indirectly affect the matter, as technology evolves, and new practices are developed Curtain will need to evolve as well.

These features would be triaged in these steps:

1. Brought up to the development team (during a sprint);
2. discussed about in a meeting with the development team, if the addition, modification, or removal could be done without issues if needed in addition to the sprint; planned approach on how to implement it;
3. implemented and tested in a development branch;
4. the development branch gets pushed up to the main branch;
5. automated test(s) is written if possible, in a separate testing branch;
6. and finally, a log would be released about the new release on Curtain.

The issue with this approach is that this approach would be slow, there would likely be numerous other features or requests. So, the request(s) or feature(s) are unlikely to be done in a singular release. These additions would likely be an update with other features which would be batch updated or developed. This approach also assumes that no additional research is needed, that the update can be done, and that it will not break the current instance of Curtain. Some updates may not be possible without breaking some inherent functionality that our application provides.

The approval process with the team would probably be talked or mentioned during a meeting in a sprint or a supplemental meeting with the sprint. The features and updates would be brought up about, talked about how to approach the modification (if possible), then how much research would need to be done (if needed).

Regarding how we would identify levels of criticality, the change would have one of three levels: immediate/high priority, medium priority, low priority.

- High priority would be assigned to include things which are currently breaking Curtain's intended purpose or a tool that is not working as it should be working.
- Medium priority would be assigned to tasks which do not need to be immediately done but will require some additional research or time.
- Low priority would be assigned to tasks which are easy to fix and do not require additional research.

Depending on the severity, the issue would be communicated to stakeholders and users. If Curtain had unintentionally not upheld the initial goals of improving a user's security and privacy then there would be a form of communication, in terms of a windows popup apologizing and informing the user about the situation. This is to ensure that users see the concern, if they do not regularly look at update(s) or log(s). Stakeholders and advisors would be informed via email about these issues. Medium or lower severities would be an update on a developer log following Curtain's development.

Budget:

This section details the cost and fiscal responsibilities which are associated with Curtain. This section aims to help individuals who have a curiosity about the expenses incurred while developing, implementing, and maintaining Curtain. When we were calculating the total budget for Curtain, we considered labor, software, hardware, and miscellaneous incurred costs.

The table below, *Table 13: Budget Table*, outlines the current estimated budget for Curtain. Most of the tools aside from the server that was used in development was free to use. To calculate the simulated labor the number of total weeks of Senior Design (32 weeks) was taken in addition to the time used to come up for the project. Each team member estimated their rough amount of time that was contributed to working on the project during the week. This averaged around 2-4 hours a week (3 roughly). Multiplying that amount in total we obtained 96 hours per person of estimated labor. The estimated labor hours were then respectably multiplied by the hourly cost was calculated by going to Levels.fyi[11][12][13], placing a few filters on our respective roles, we got ~$46 ,~$86, , and ~$33 so we used that for the hourly rate. This is detailed in the Labor section of *Table 13: Budget Table*.

Our software and hardware costs, included in some cases pre-existing hardware (laptops, desktops, monitors, microphones, and speakers). These materials were used for development and meeting virtually. We used the following items with their respective prices for development and meeting virtually when we could not physically meet up. These items were also included under the Hardware- External and Software-External description in the *Table 13: Budget Table*.

Other hardware costs included wires and adapters which were used to connect from our respective computers, laptops, microphones, etc. These items were also included under the Hardware-External description in the *Table 13: Budget Table*.

We also assumed that because of the varying WiFi rates that we had to look at the average rate of WiFi in the Cincinnati area, which was found to be ~$46 a month from looking at U.S. News[14] (obtained by adding the monthly costs of all the providers and dividing that amount by the providers). Since, WiFi requires both hardware and software this cost was included in the miscellaneous costs section of *Table 13: Budget Table*. Phone plans were not considered a portion of the budget, because we did calls through web conferences. Furthermore because of the lack of an office space and physical meetups, we did not include the costs of renting an office space in the Cincinnati area.

[11] Levels.Fyi, Last modified 2022, https://www.levels.fyi/comp.html?track=Software%20Engineer&search=cincinnati&yoestart=0&yoeend=1&yoeradio=New%20Grad.

[12] https://www.levels.fyi/comp.html?track=Business%20Analyst&search=cincinnati&yoestart=0&yoeend=1&yoeradio=New%20Grad

[13] https://www.levels.fyi/comp.html?track=Software%20Engineering%20Manager&search=cincinnati

[14] Rob Pegoraro, "Internet Providers In Cincinnati", U.S News, Last modified 2022, https://www.usnews.com/360-reviews/internet-providers/local/cincinnati-oh.

Table 13: Budget Table

		Rate Per/Hr	Hours or Units	1 X Costs	Ongoing Annual Rate Per/Hr	Work Effort (Hours)	1 X Support Cost
Labor	Developer	46	192	$ 8,832	46	288	$ 8,832
	Management	86	96	$ 8,256	86	96	$ 8,256
	Analyst	33	96	$ 3,168	33	96	$ 3,168
						Total	$ 20,256
Software - External	Server		1	$ 534.41			$ 534.41
						Total	$ 534.41
Hardware - External	Laptops		4	$ 11,526			$ 11,526
	Desktops		2	$ 4,800			$ 4,800
	Monitors		7	$ 2,030			$ 2,030
	Microphones		4	$ 515			$ 515
	Keyboards		4	$ 378			$ 378
	Mice		4	$ 84			$ 84
	Phones		4	$ 1,319			$ 1,319
	Wires		9	$ 138			$ 138
	HBA Card		1	$ 120			$ 120
						Total	$ 20,910
Misc.	Wifi			$ 368			$ 368
						Total	$ 368
TOTAL							$ 42,068.41

Problems Encountered and Analysis of Problems Solved:

- Initial project approval

Our initial project was rejected, which was going to be a decentralized mental health platform using AI to help provide accessible mental health. Upon consulting a few faculty advisors, they pointed out our project was not only ambitious. But there were too many unanswered questions for securing and using the technical stack that we originally wanted to proceed with. For example, using sentience to determine how a user's input would be processed. There were also too many unknowns with our initial project proposal.

We solved this by taking our initial criticism into consideration and decided to move forward with another concept or idea for our project. We had several brain storming sessions, but ironically the project we went with was not an idea that we threw out during our brainstorming session. But it was when one of our team members, November wanted there to be some application to auto reject or use the minimum number of cookies. We took that idea and the feedback we received on earlier ideas, and we conceived the idea for Curtain, focusing on improving the end user's privacy and data control.

- Miscommunication with Team

There was initially a lot of miscommunication between the different team members, e.g., there was a presumed assumption initially on how others would proceed from the project. This led to a lot of confusion on where to start for others and for some a thought that no progress needed to be done. This was quickly addressed after our first standup, when others talked about how they approached projects and development. We talked about specific skillsets which we had and some weaknesses, e.g., some people had more web development experience than others and others were not remotely aware of how to develop.

This problem is not necessarily solved, as we still have some miscommunication and misunderstandings from attempting to communicate progress and help requests. While having standups did help reduce concerns and keep people up to date, there was sometimes a need for additional support or further elaboration outside of stand up. We also had a Kanban board which was used to further support our communication attempts and convey information to those who needed help. We suspect we will still have communication issues based on our vastly different backgrounds and knowledge. Regardless, it is not possible to have perfect communication, but our communication has improved since our first time working together.

- Developmental issues/Progress

As mentioned above, there was a lot of miscommunication initially. While progress and development increased as we continued to develop as both a team and on the project. There were a lot of developmental issues we ran into, first off none of us had previously developed a web plugin tool. Secondly, we made a lot of initial assumptions about how browsers ran which was incorrect. Third, in some developmental cases there were issues with the pipeline overwriting pre-existing code. Fourth, figuring out how to solve our storage issue.

Part of the issue was a slightly resolved by initial research, which was done by our team lead, Kai, who conveyed information to the rest of the team. This made developmental progress a lot smoother as he did a great job of providing concise and accurate information. However, this slightly hindered his developmental progress which he was hoping to implement into Curtain. We also had quite a few personal events which interfered with the initial development of Curtain, e.g., other classes, personal circumstances, etc. This portion was communicated in order to help,

not stunt, progress; however, some conflicts were unavoidable and resulted in development delays.

We also discovered that there was going to be platforming development issues, e.g., Gecko's plugin infrastructure and backend code is vastly different from Chromium based ones. We found the discrepancies were too large and we would need to scale back our development to not include Firefox's browser at this time because of the great discrepancy. Which saved us time; however, we would have loved to included Firefox as a browser that can use our plugin as well.

There were other developmental issues in terms of figuring out the storage. We initially had a localized instance of our blocklist for our AdBlocker tool. However, we soon realized that this would not be sustainable for the long-term, as locally stored lists become outdated as quickly as advertisement technology evolves and making one of our tools ineffective. We had some difficulty figuring out how to store our incremental data, which stemmed from the question of if it would best to store the data at a localized instance or in a server. We concluded that for specific user data it would be best to have a localized instance of the data, i.e., assuming that the user had consented for logging where then the plugin would display the numbers of ads blocked and cookies deleted.

- Time Constraints

We found the biggest challenge of all was the time constraints put on our project Curtain. This sometimes led to unpolished and unoptimized code being pushed up to our main branch to attempt to meet some of our initial goals. Furthermore, as mentioned above, no one on the team had experience developing a web plugin before. So, there was hinderance to our initial development on the project, it did not help when we had personal conflicts with the project as well.

Additionally, as mentioned above we had to scale back some of our developments because of personal conflicts, conflicts with backend and infrastructure in browsers, and our initial slow start. Had we been given more time; we would have been able to pursue these developments. However, it seems as if that is unlikely to be developed with the time constraints given.

Conclusion

This section will detail the lessons learned in the fall, spring, skills, findings, and a self-reflection component. In conclusion, the project development was a success as we developed an application, learned how to work as a team, and developed skills which have positively impacted all of us.

Findings

Overall, our findings were that there is this power imbalance in the online sphere when it comes to privacy. The original concern in the online sphere was gathering and storing data, albeit this concern has shifted into a moral dilemma and concern because of the amount of information being gathered on a person is often not consensual or not understood. Tracking online primarily occurs in two different contexts within-site and cross-site tracking. Some of the tracking technologies has expanded to start upon starting up one's browser. Our application Curtain has made it possible for people to take back control over their data. Curtain not only provides individuals with the ability to control their data being gathered on them, but also the control to whitelist some websites and ability to turn on or off some tools which they do not deem necessary to use. Curtain is a web-plugin which centralized data privacy control tools, this was possible with blocking advertisement, deleting cookies, reducing tracking.

Fall Lessons Learned

- Assume that you will take more time than initially thought

  It is generally better to be done earlier or later with functional working code that is good as opposed to rushing incomplete and unpolished code, which leads to a lot of headaches in the future.

- Overcommunication is better than under communication

  During our time working together we got relatively close with one another. Especially with sharing personal conflicts which made it not possible to complete or work on the project. Being aware of the situation made it possible to where we were able to either plan around the issue or have someone work in an area where that person was working with or on.

  During testing, we found that overcommunicating test results and procedure often led to being able to troubleshoot more efficiently and replicating the issue initially observed.

- More details or specific features are not always for the better

  Our team learned that we needed to focus less on specific details and more on the bigger picture. We had a lot of initial plans to make the plugin widely accessible to users and the different tools we wanted to implement. During our time developing this project, we asserted that it would not be possible to develop for both the Gecko and the Chromium browser engines at the same time. We also found that when we were originally developing or planning to develop different features this led to a lot of disanalogous understanding and code.

  We also found that there was often a lot of times where different codebases and tools would conflict with one another. So, we would have to remove some functionality to make the tool work properly.

- Tests should be conducted in different environments

  As mentioned in the testing section, we noted that people who did not contribute to developing a particular tool or feature were more likely to find bugs. This is likely because the developer has an idea on how it should be run to perform the tool's intended function; however, an individual who did not develop on it would not necessarily intuitively follow the procedure the developer did to have the tool run.

Spring Lessons Learned

- Testing should continue to happen periodically

  This makes it possible to find undiscovered bugs as well as ensuring that our application is running as expected. We noticed that there were some abnormal behaviors outside of testing, which made us realize that some of the tools we had created did not operate as expected.

- Storage can be a difficult thing to tackle

  We struggled with deciding where to store, place, and access the data. This also included the associated pipeline, including moving data sources between the client and server architecture we had developed. We had discussed the mechanics and researched to what extent storing the data locally as opposed to remotely. We concluded that the storage would be dependent on the tool case and need. This made it even harder to develop for because our architecture had become more advanced than what we had initially planned for.

- Scope will change dynamically

  This is essentially what was learned last semester, we were still ambitious with some of what we wanted to accomplish in terms of capabilities for our project, Curtain. However, even our reduced goals for our project we found that we still needed to reduce our scope to meet the required time constraint while also accomplishing what we wanted to accomplish.

  As we worked on the dashboard, we decided to combine the dashboard and the settings screen into one view, which was much faster to develop and test. This also made it more accessible for users to use our application as they did not have to worry about additional settings and views. This also made for a much more polished product in terms of the dashboard, being able to work and accessible as intended.

Skills and Abilities Developed

- Communication

  Prior to working together, we were not all well acquainted with one another nor were we familiar with each other's work dynamic. For example, some members of the team said they operated best when there was a detailed checklist given in task in the Kanban board. Other members worked fine without a detailed checklist.

- Practices for plugin development

  We learned how to develop for web plugins and the better practices for doing so. There were different forms of debugging platforms and compression which we were not originally used to nor knew. Overtime of developing for the plugin we learned better practices for developing.

- Estimating Abilities

  Prior to working on this project together, we had some rough idea of how to estimate. However, most of the initial estimations were incorrect. Upon continuing to work together as a group and learning about one another's circumstances. We became better at roughly estimating our tasks completion as well as when our other team members would need some assistance without needing to directly communicate this to the team.

- Teamwork

  With working with one another, since we had gained better communication, estimating abilities, and practices. We had become better at working as a team together. This led to much faster productivity in terms of the developmental cycle, testing, documentation, and supporting one another.

Self-Reflection

The developmental progress of this project was difficult. We had many challenges and sometimes did not have a solution to the immediate problem. From these challenges we learned a lot of important lessons outlined in the fall and spring section. Most of the problems we found tended to be with the scope of the project and communication issues. Since working on this project, we have gotten better at communicating with one another. As well as learning to specify and lessen the scope of what magnitude of tools may be implemented to still have a functional project. If we were to attempt to do this project again from scratch, we would implement a few practices initially such as: establishing communication guidelines for each individual (there was a lot of initial misunderstandings because of a lack of understanding how each individual communicated), more regular communication by fostering a healthy environment to vocalize criticism, additional accommodations provided earlier by vocalizing issues immediately, a more formalized way of communicating, sliding roles of researcher, and more soft deadlines to help keep some people on track.

References

Aiden, Erez, and Jean-Baptiste Michel. Uncharted. 1st ed. New York: Riverhead Books, 2013.

Furubotn, Eirik, and Svetozar Pejovich. "Property Rights And Economic Theory: A Survey Of Recent Literature". *Journal Of Economic Literature* 10, no. 4 (1972): 1137-1162. https://www.jstor.org/stable/2721541.

Kalavri, Vasiliki, Jeremy Blackburn, Matteo Varvello, and Konstantina Papagiannaki. "Like A Pack Of Wolves: Community Structure Of Web Trackers". *PAM 2016: Passive And Active Measurement* 9631 (2022): 42-54. doi:https://doi.org/10.1007/978-3-319-30505-9_4.

Lanier, Jaron. Who Owns The Future?. New York: Simon & Schuster, 2013.

Leber, Jessica. "A Dollar For Your Data". MIT Technology Review, Last modified 2022. https://www.technologyreview.com/2012/06/08/185769/a-dollar-for-your-data/.

Levels.Fyi, Last modified 2022. https://www.levels.fyi/comp.html?track=Software%20Engineer&search=cincinnati&yoestart=0&yoeend=1&yoeradio=New%20Grad.

Levels.fyi, Last modified 2022. https://www.levels.fyi/comp.html?track=Business%20Analyst&search=cincinnati&yoestart=0&yoeend=1&yoeradio=New%20Grad

Levels.fyi, Last modified 2022. https://www.levels.fyi/comp.html?track=Software%20Engineering%20Manager&search=cincinnati

Peacock, Sylvia. "How Web Tracking Changes User Agency In The Age Of Big Data: The Used User". *Big Data & Society* 1, no. 2 (2014). doi:10.1177/2053951714564228.

Roesner, Franziska, Tadayoshi Kohno, and David Wetherall. ") Detecting And Defending Against Third-Party Tracking On The Web". *9th USENIX Symposium On Networked Systems Design And Implementation* 12: 155-168.

Pegoraro, Rob. "Internet Providers In Cincinnati". U.S News, Last modified 2022. https://www.usnews.com/360-reviews/internet-providers/local/cincinnati-oh.