Trailmix

Project Part 4: Project Prototype

Team 8

Sarah Cooper

Matthew Deagen

Leo Galang

Marc Ace Montesa

Instructors

Sergiu Dascalu, Devrin Lee

External Advisors

Jay Thom, Nathan Thom, Maxwell Thom

Department of Computer Science & Engineering

University of Nevada, Reno

December 11, 2020

Table of Contents

Abstract

As large tech corporations like Google and Facebook continue to dominate the Internet through targeted ads, privacy becomes a growing concern. To address this issue, the team will build a Google Chrome extension called Trailmix that randomizes the user's browsing history to allow for complete anonymity and reduced advertisement tracking. Conceptually, one user's history can be viewed as the peanuts, another's the raisins, and a third user's as the chocolate. When Trailmix is enabled, it will scramble their data together and the user is left with a random assortment of browsing history. To give users another layer of control, a secondary functionality will let them force specific browsing history to be applied through the use of keywords. On the technical side, Trailmix will be written in JavaScript React and will utilize a database system that allows for users to have complete functionality, while still allowing the user to actively mitigate and control their preferences with forced browsing history.

Introduction

With the rapid advancement of technology in the past two decades, privacy has gone unregulated. Corporations such as Google and Facebook use personal data to build individual profiles that let advertisers target groups based on their interests. Although users may agree to this when they check off the terms and conditions, many are largely unaware of this practice. Technology is advanced enough to predict our behavior and may be able to manipulate it in the near future. To help remediate this issue, the team will create a browser extension that randomizes user browsing history in an effort to promote privacy. The goal of this project is to allow users to be anonymous and prevent their data from being misused by large corporations.

Since the previous assignment, there have been a couple minor changes. The Trailmix team discussed what the user preferences are and decided that the browsing history will be the data to be collected. Cookies and logs might also play a role into the browser extension, as they contain data used by third parties to target ads. Trailmix will not have to cater to individual websites since browsing history is more of a universal point of extraction. Another change made was the implementation of a whitelist; users will be able to force browsing history through keywords. Previously thought as a difficult task, the team will implement this feature if time allows. This will give users more options to be anonymous as well as provide uniqueness to the Trailmix application.

Prototype Objectives & Functionality

The three main objectives when making our prototype was to implement the prototype in a user interface design which is easy to visualize, to implement a functionally usable program that would follow the UI design [1], and to implement a few security features into our prototype. One of the Trailmix team's goals for this product is to provide anonymity and security for a user against targeted advertisements and to ensure security was considered in the design of the product. The team started the prototype design with a meeting on the functionality to include in the prototype that would best showcase their idea of how the final product will look and act like. After figuring out the functionality, the group decided that two members working on the user interface and two working on the code would be most efficient. However, to ensure that no one member was overworking or needed help, all members would be in a voice call and work together.

The functionality for the prototype was coded in Python, however the final product will be in JavaScript using the React framework. There are 12 functional aspects of the prototype code and one from the user design interface: *Sign In*, *Logout*, *Create Account*, *Enable/Disable* functionality, *Refresh*, *Show Preferences*, *Add Preferences*, *Show Logs*, *Clear Logs*, *Add to Logs*, *Enable/Disable* sending data to Trailmix and *Change Password*. The *Sign In* function allows a user to input their credentials to get into their account. *Logout* simply logs a user out of their account and disables all functionality including search data being sent to Trailmix. *Create Account* is for when the user does not have an existing account and would like to use Trailmix. This function requires a Username which should not be the user's real name. If a user tries to enter a username that is already in the database, Trailmix will deny them that username and ask for a different one. This is because Trailmix uses the user's usernames as a primary key in the user database. However, this function will likely change to a set *UserID* number randomly given to a user in the final product. Lastly, the user will add a password to their account for authentication and security. The password is required to have more than 10 characters long, and for the final product will require the use of special characters as well as upper and lower case and numbers. This is to ensure that the password would take a significantly long time to brute force. Trailmix also hashes all of the passwords with SHA-256 and a randomized salt, which is also hashed. This is to ensure that if the database were to be accessed, the passwords would be secure.

The next set of functions pertains to the randomized preferences which is the main functionality of our prototype. The *Refresh* function allows for the user to reset their randomized preferences whenever they want, along with the set time of 24 hours. The *Enable* and *Disable* function allows the user to turn off or back on the functionality of Trailmix. This does not stop Trailmix from

collecting data, however there is a function that can also be enabled to stop all data from being sent. As well when the functionality is re-enabled, Trailmix will refresh the randomized preferences. The *Show Preferences* functionality simply lists the current randomized preferences for the user as well as any preferences they have added to their list. They can *Add Preferences* to their list which is a preliminary version of the whitelist idea. This will allow users to type in or choose certain preferences they want to see recommendations for. This functionality in the prototype allows users to input certain preferences that will be included in the random list that is "displayed" to Google. In the prototype when *Refresh* is chosen, the added preferences disappear, but the team hopes to implement a permanent version in the final product.

The last two sets of functions are the *User Log Settings* and general *Settings*. The user log setting of the code prototype is the same as the *User Preferences* list on the UI design. This list of functions includes *Show Logs*, *Add to Logs* and *Clear Logs*. *Show Logs* function is to show a user the information that Trailmix is collecting from them. Trailmix only collects the users search history as string literals, such as "dogs," "helicopters," "Oculus Rift S price," etc. Then there is the ability to clear these logs for added privacy. Last is the *Add to Logs* function, which will be later removed in the final product. This functionality is to showcase in the prototype how when a person searches anything on Google, the information is stored and can be sent to Trailmix to add to the list of preferences that anyone can get. For the general *Settings* there are two functions, one implemented in the code and UI design and one in the UI design alone. for the UI part of the prototype, there is a change password, however that was not implemented in the code. This was not coded because it was not a main function that showcases how Trailmix is different and useful. This last functionality is the ability to check or uncheck if the user wants their data sent to Trailmix. This is very important because it allows the user to continue to use the extensions without forcing the user to give Trailmix their data. For security and anonymity reasons, the team opted to include this.

[1] Link to Figma UI prototype: https://www.figma.com/file/CIc6MvZfrwwhkJREzv9Cng/UI?node-id=0%3A1

Prototype Development

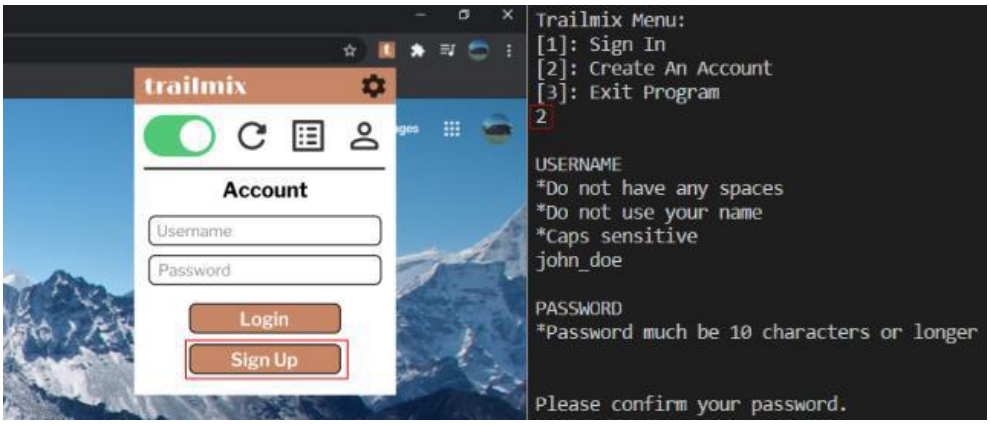

Figure 1: Visual and functional representation of the create account functionality. This is the first page a new user will see when they click the extension icon. The rules for creating the username and password are specified and must also enter the password twice to make sure that they match. There will be an error message shown if the entered passwords do not match.

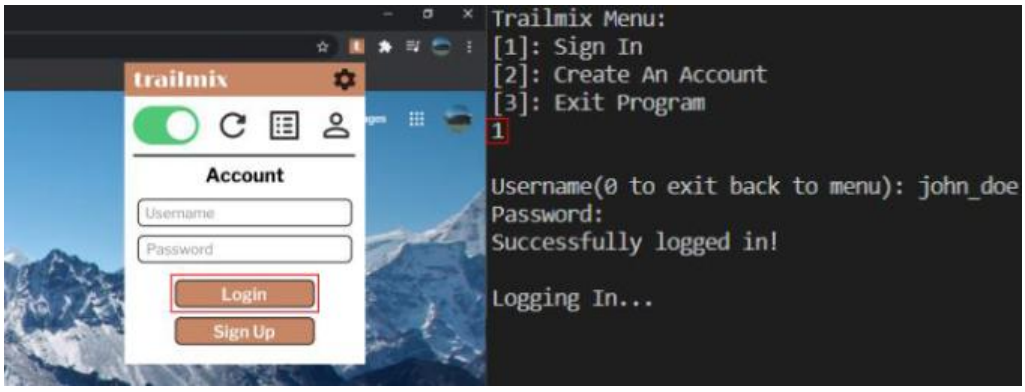

Figure 2: Visual and functional representation of the login functionality. Each password is hashed with a salt value unique to the user. This ensures that if two users have the same password, they will have different hash values of their password.

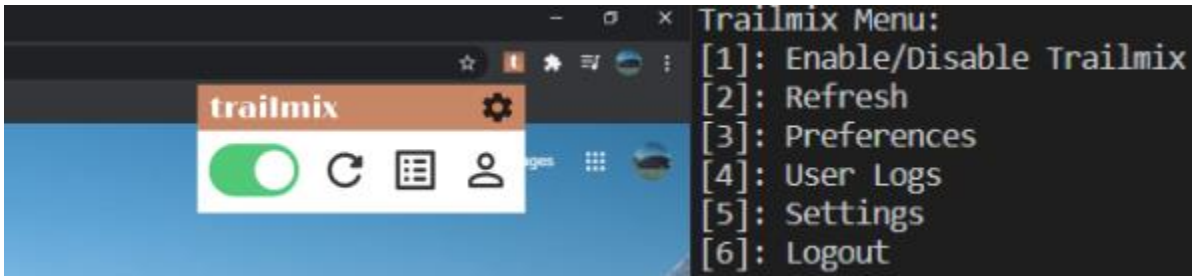

Figure 3: Visual and functional representation of the main menu. This is what first appears once a user has clicked the extension icon when they are logged in.

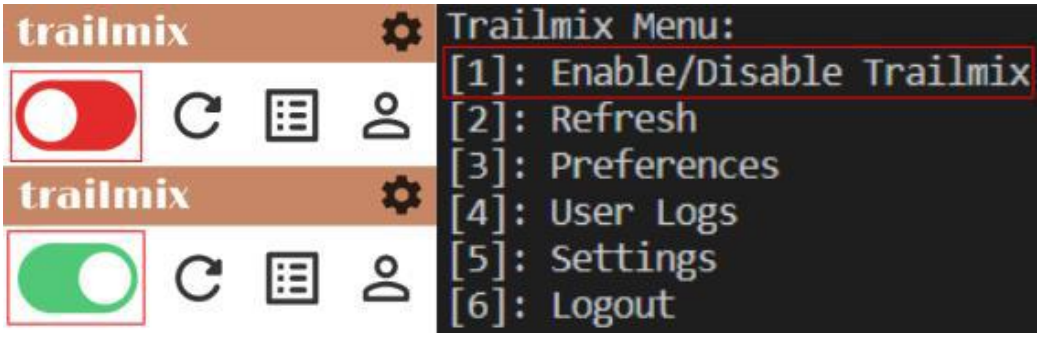

Figure 4: Visual and functional representation of the enable/disable switch. This part turns the extension on and off. The current preferences will also be refreshed when the extension has been re-enabled.

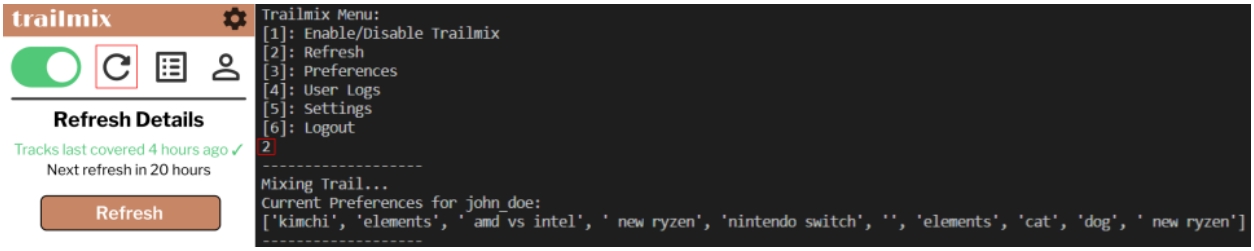

Figure 5: The visual and functional representation of the refresh component. When refreshed, the current preferences will be updated with a randomized list of preferences.

Figure 6: The visual representation of the current preferences and user preferences. The user preferences are based on the account owner, while the current preferences are randomly generated from a list of all Trailmix user's preferences. This provides a "mask" to hide the user's real preferences when browsing.

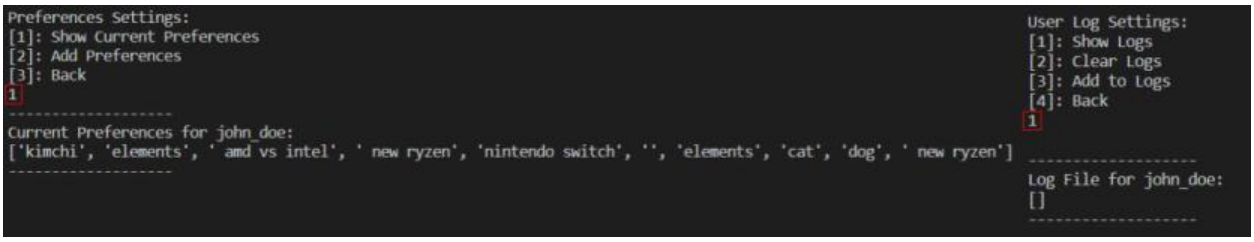

Figure 7: The functional representations of the menus associated with the current preferences and the user preferences (or user logs). Because this account is new, there are no user logs stored. The current preferences are based on other Trailmix users.

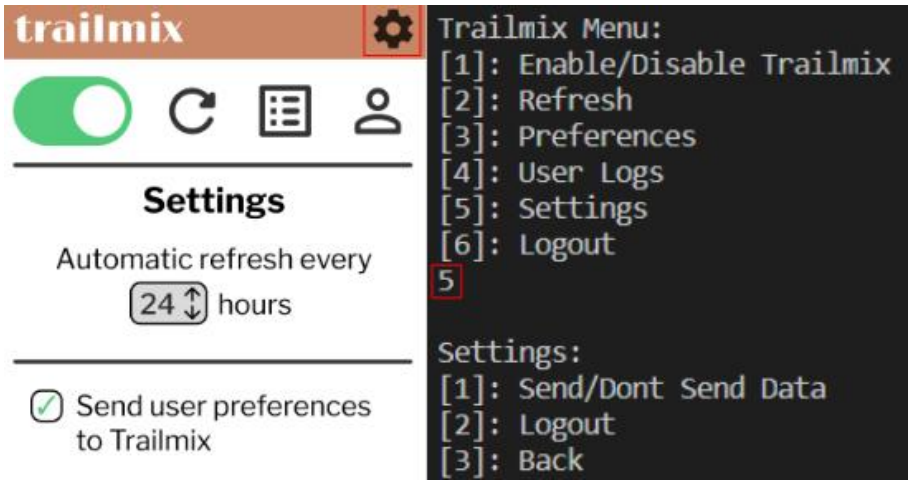

Figure 8: The visual and functional representation of the settings menu. The extension can be set to refresh the current preferences after a specific amount of time, and there is also the option to send or not send user preferences to Trailmix databases.

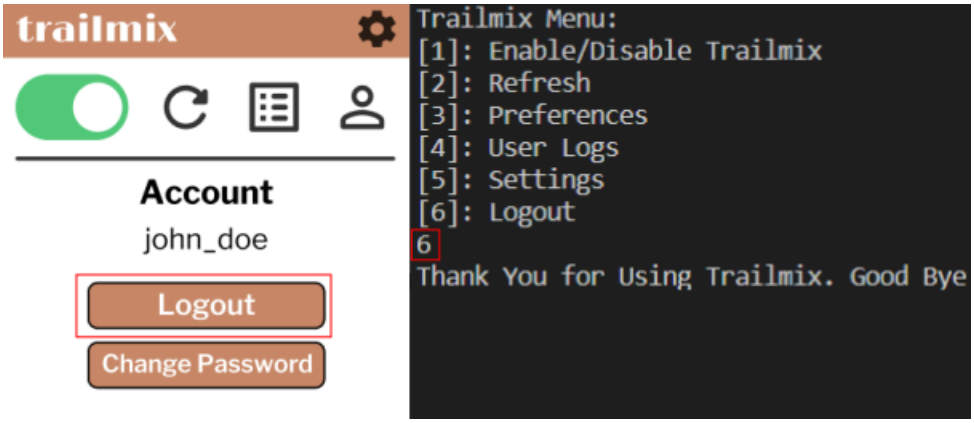

Figure 9: Visual and functional representation of the logout component. Once the user logs out, the Trailmix window will close. If the user then clicks the extension icon, they will see the login screen (see Figure 1).

Evaluation of Prototype

The Trailmix team held a user demonstration and feedback session with their advisors, Nathan and Maxwell Thom, prior to meeting with their designated instructor; Jay Thom unfortunately could not make this meeting. The meeting with the two provided great feedback in how the team should approach the technical design later in the development. In this, they liked the concept and design of the user interface, commenting on the intuitiveness of it. The two had also expressed a great interest in the terminal demo, commenting on the security provided by the encryption used during account creation. They had recommended the necessity of special and capital characters to help bolster security. Lastly, they had expressed concern on how Trailmix will utilize the given profile/preferences to confuse the search algorithms. In this Maxwell suggested Selenium, an automation framework that will run in the background of the user's browser to help obfuscate cached data. Overall, the two advisors present provided valuable suggestions for the Trailmix team.

Demo Prototype

Our prototype demo was on December 8th, 2020 at 2:00pm. The team's meeting with Dr. Dascalu provided a good insight into the future expectations for the project. Overall, he stated that the demo showcased quality and simplicity in the design and implementation of the user interface. He did express the risk with how a simple design can appear to the team's peers, that while he understands much of the complexity can lie in the back-end design, a way to demonstrate more of that complexity from the back-end to the end-user may help remedy this. Other than that, Dr. Dascalu also stated that the use of "current" preferences is often misleading, and rather uninspiring. He made some suggestions for substitutes that the team will consider, including curtain, mask, disguise, etc. The feedback obtained from Dr. Dascalu proves valuable to the Trailmix team, and will be put under much consideration going forward with the software design.

Changes Needed to Software

The prototype was found to be a successful milestone for the project's development; however, much of the technical functionality still needs to be worked on. For starters, the team plans on transitioning from a text-based database, to an SQL-based database. This is to provide added security to the user information should they choose to share it with the developers. In the same vein, the team wishes to transition from literal usernames to a *UserID* system to navigate through the database; again, as an effort to address concerns for user privacy. Another security feature the team plans to include, based on the feedback from the external advisors is, better specifications for creating usernames and passwords to increase security. The next planned implementation is to integrate the UI components with the Terminal functions. Ideally, this would

be done using JavaScript React as the team further develops the concept. Another item the team expresses interest in is the use of Selenium, to aid in Trailmix's obfuscation. Lastly, based on the feedback from the instructor, the team will further explore the possibilities and risks associated with a simplistic interface. Overall, the data structure and the software specifications remains consistent with the restructured design the team proposed later in the development. The use of hardware may become apparent as the development continues, with the necessity of physical server space being a possibility.

Team Contributions

Sarah Cooper worked with Marc on the Python demo. She did the main function as well as the preferences and log functions which took about a week or 8-10 working hours. In addition she wrote up the prototype objectives and functionality part of the written report which took about 2.5 hours total. She also made contributions to the demo on the security functions.

Matthew Deagen worked with Leo on developing the visual prototype using Figma. He also worked on the screenshots and descriptions for Section five of this assignment. About eight to ten hours were spent on developing the visual prototype and about one and a half hours were spent on section five.

Leo Galang worked with Matthew on the Figma visual prototype as well as the abstract and intro, taking about 8-10 hours to complete. Leo also presented the introduction and visual demo of Figma to Dr. Dascalu.

Marc Ace Montesa had worked with Sarah on the Python demo. He had specifically worked on the password encryption and obfuscation functionality. In addition to that, he, along with Leo, had also presented the demo to the advisors and Dr. Dascalu; with support and clarifications from both Sarah and Matthew. Lastly, he had written sections 6, 7, and 8. The time spent ended to about two hours for the written portion, with roughly a week (8-10 working hours) developing the prototype.