GNG<1103/2101>

Design Project USer and Product Manual

The Point Marketplace

Submitted by:

Point Marketplace Group 1

Nina Blaney, 300234634

May Danhash, 300229880

Fadel Aameh, 300249346

Dane Kontic, 300234111

Mason Chopik, 300242710

November 27, 2021

University of Ottawa

Table of Contents

List of Figures

List of Tables

List of Acronyms and Glossary

Table 1. Acronyms

Acronym	Definition
BMO	Bank of Montreal
CAD	Canadian Dollar
CIBC	Canadian Imperial Bank of Commerce
CRX	the Learning Crossroads Building
RBC	Royal Bank of Canada
TD	Toronto-Dominion Bank
ToS	Terms of Service

1. Introduction

In this manual we will go through how our final product works, getting started, using the system, troubleshooting and support, product documentation, conclusion, and recommendations for future work. In the beginning of the project, we discovered the lack of integration of different loyalty programs. To add, users were unable to trade, buy, and sell points in one place. During the beginning stages of our research, we identified certain factors that could cause problems when trying to convert a certain loyalty point to another. Therefore, we found a solution that will be explained throughout the user manual. In summation, the idea that we came up with to democratize and integrate other loyalty programs was to make a point marketplace where users can sell, buy, and trade points. Our website has a front and back end that is secured by Vue.js 3 and Fireplace. Only users signed up and verified are able to log into the website and use the website.

2. Overview

In the world of loyalty programs, people tend to be a part of multiple different systems, collecting points as they go. However, nowadays, most of the general public is part of more than one program. Due to this fact, points collected by the user for certain programs just accumulate in the user's account. This overall accumulation is an issue as it blocks the user from gaining the various benefits, such as saving money on big purchases.

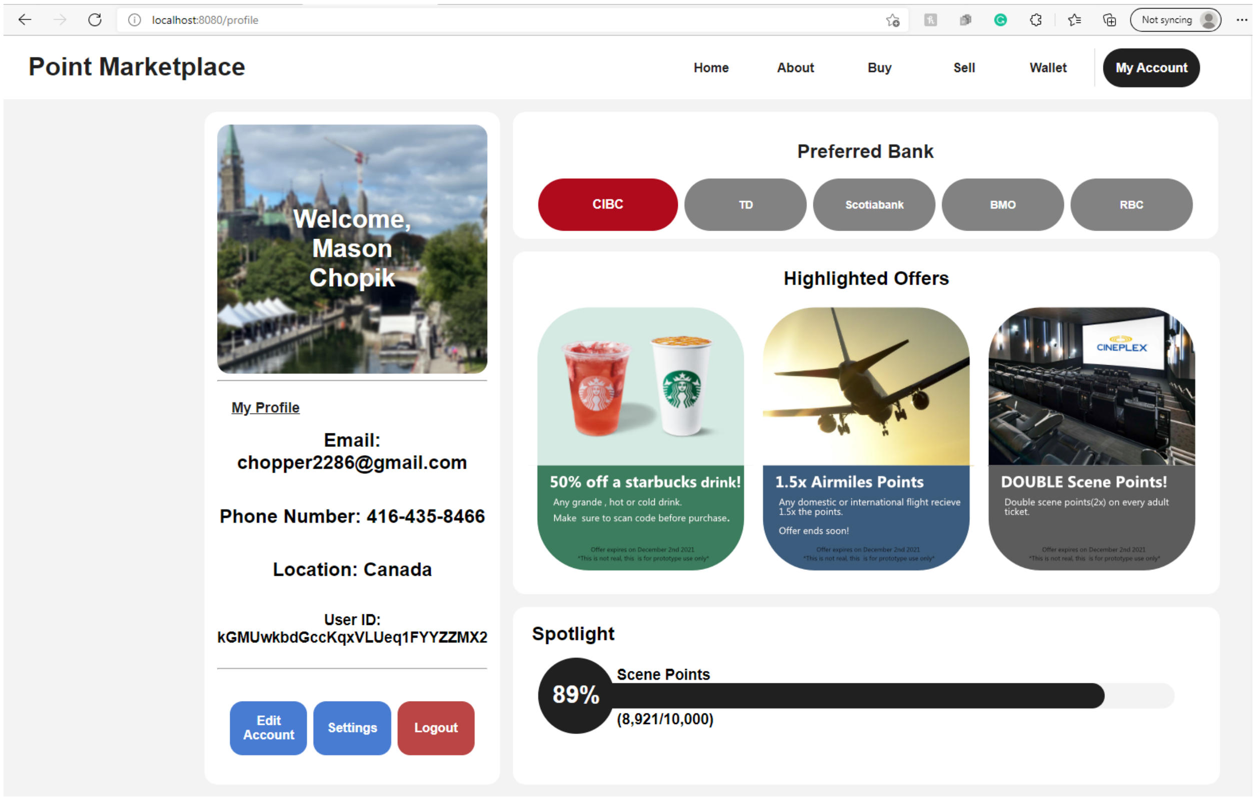
A simple solution to this problem would create a way that can convert points between loyalty programs. However, an issue arises in the fact that for most loyalty systems, a conversion between points goes against their TOS. A way around this important issue is to introduce a middle man; a person or place that the public could sell their points to, gaining money in return, and then, in turn, be allowed to purchase the points that they desire from a different, or even same, person or place.

With this in mind, a solution presents itself. A point marketplace system, with the various pre-existing banks being the middlemen. With this solution, both parties, the bank and the costumers of both the bank and loyalty programs, gain revenue, allowing for those points that are just being accumulated to finally be put to use.

This solution differentiates from other products as no other application, besides the existing stock market that only deals with investments for public companies, exists. Additionally, with the solution, the security of the application would be highly prioritized, making sure that every user's data is safely stored. Furthermore, the solution would be user-friendly, allowing anyone to use it with ease.

2.1 Final Product

The final product, called the Point Marketplace, met what was the goal that was initially set out, which was to make a solution that was secure, and easy to use. Even though the final product is not in full functionality, more legal work is required to make a fully functioning stock market-type system. However, the group nonetheless is very excited, and proud of the work that has been done to reach this point.



Email: chopper2286@gmail.com

Phone Number: 416-435-8466

Location: Canada

User ID: kGMUwkbdGccKqxVLUeq1FYYZZMX2

50% off a starbucks drink!
Any grande , hot or cold drink.
Make sure to scan code before purchase.
Offer expires on December 2nd 2021
This is not real, this is for prototype use only

1.5x Airmiles Points
Any domestic or international flight recieve 1.5x the points.
Offer ends soon!
Offer expires on December 2nd 2021
This is not real, this is for prototype use only

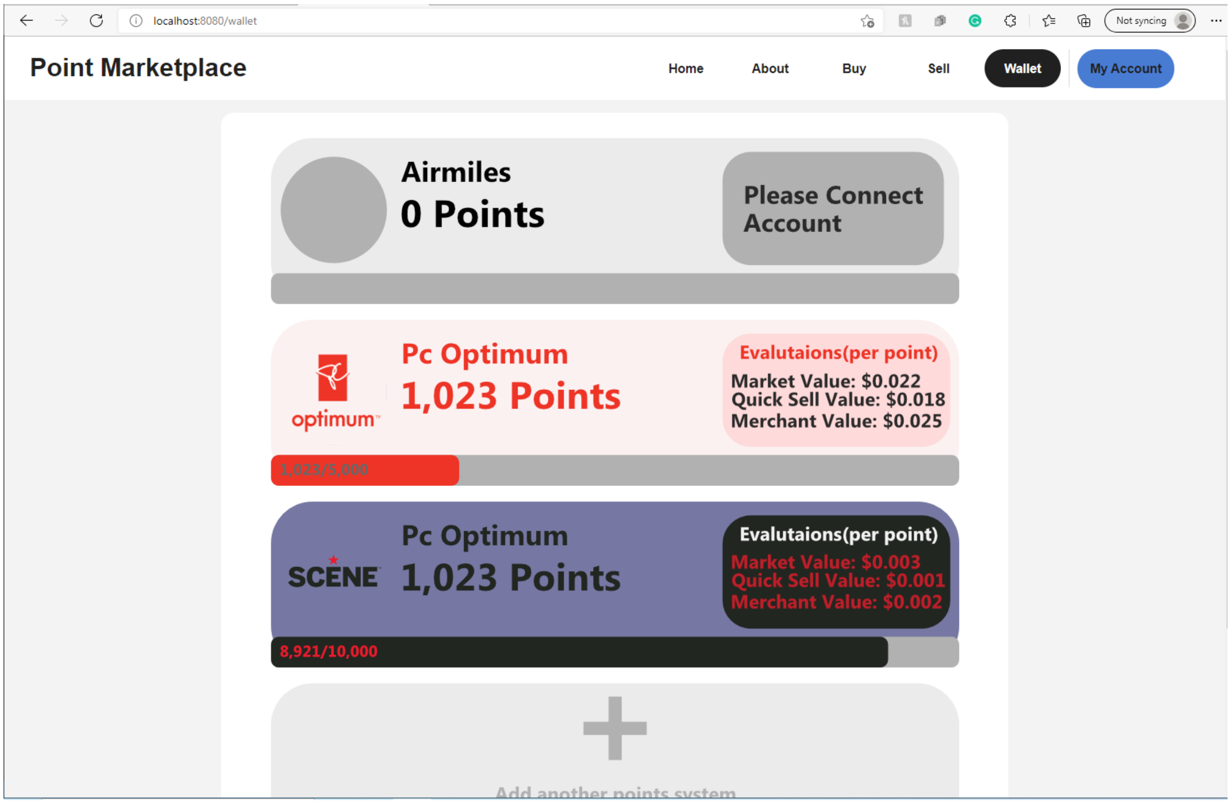
DOUBLE Scene Points!
Double scene points(2x) on every adult ticket.
Offer expires on December 2nd 2021
This is not real, this is for prototype use only

Scene Points 89% (8,921/10,000)

Figure 1:Profile Page

Description:

This is an example of the user’s account page. Here, the personal information of the user is viewable by them, such as their email, phone, location, user ID, and name. Also on the account page, the user can select their preferred bank, (in the image it is CIBC), as well as see popular offers that are happening at the moment. Also, the user has the opportunity to see how many points they have, as well as have the option to edit their account, go to their settings, or log out.



Airmiles
0 Points

Pc Optimum
1,023 Points
Evalutaions(per point)
Market Value: $0.022
Quick Sell Value: $0.018
Merchant Value: $0.025
1,023/5,000

Pc Optimum
1,023 Points
Evalutaions(per point)
Market Value: $0.003
Quick Sell Value: $0.001
Merchant Value: $0.002
8,921/10,000

Figure 2: Screenshot of Wallet Page

Description:

Here, the user's wallet can be viewed. In the wallet, it contains what loyalty system they have connected to the Point Marketplace, as well as the 3 different values for the points; Market Value, Quick Sell Value, and Merchant Value.

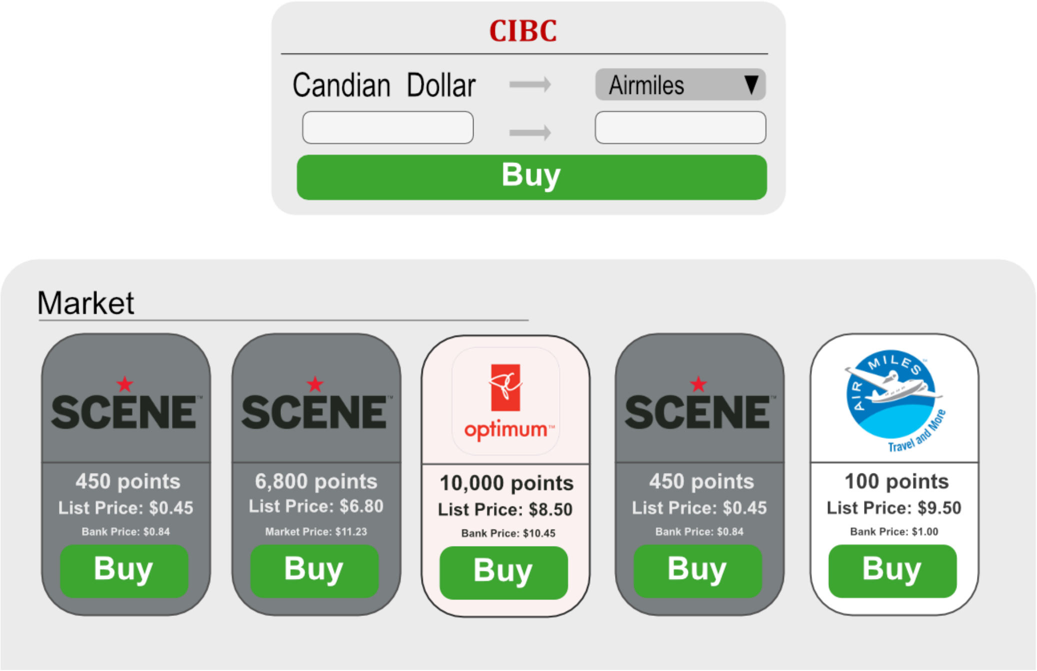
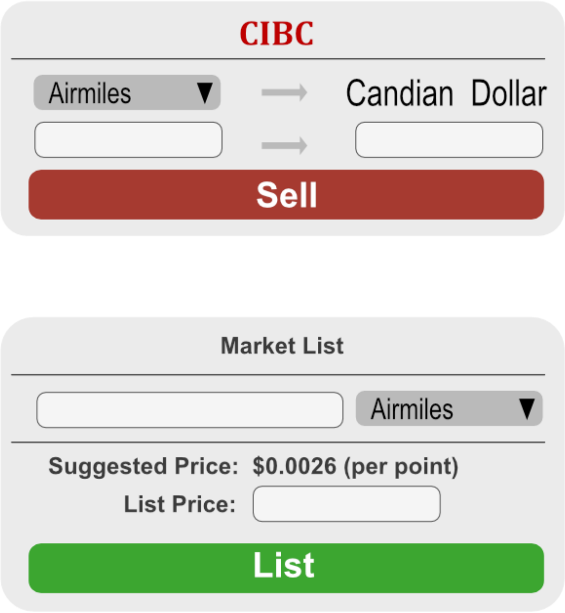


Figure 3: Example of the Points Buy feature.

Description:

This is an example view of the points buy option. Here, the amount of money can be entered, (currently in CAD), and the amount of the selected loyalty program, (in this case Air Miles), that can be bought from the user’s preferred bank. Bellow, a list of different amounts of points currently listed on the market by individuals are available to look at.



Suggested Price: $0.0026 (per point)

Figure 4: Example of the Points Sell feature.

Description:

This is an example of the points sell feature. Here, the user can input the number of points that they would like to sell, and the amount of money they would get in return. Also, if the user would not like to sell to the bank, but put their points up on the marketplace, they can list it and a suggested price per point would be given, and the option to put a list price.

2.2 Cautions & Warnings

Since this design is still in the early phases, there are no cautions or warnings to provide at this time. However, because the design is in the early phases, there are limitations. In its current state, actual buying and selling of points cannot happen, as setting up a fully functioning stock market would require more legal power than what the group currently has.

3. Getting Started

This section provides some general information on how to get started in this program.

3.1 Set-Up

To set up access to the program, you must create an account by selecting the “create an account” button. This will bring you to a create an account page and users will be prompted to create an account using their email and entering a password, location, and phone number.

3.2 User Access Considerations

To be able to create a profile for this program you must have a working email. From there, to actually use the program you must belong to a bank and have access to all relevant banking information. After inputting your bank information, you can now use the program.

3.3 Accessing the System

To access the system, be sure to remember your email and password and keep them in a secure location, as they are how you access your account. You enter your username and password in the sign in space right when you access the website.

3.4 System Organization & Navigation

The home screen is the main hub of the website where most of your account information can be found. At the top of the screen, you can access all systems; Home, Buy, Sell, Wallet, and My Account.

3.5 Exiting the System

Simply closing the tab that the program is open on is good enough to exit.

4. Using the System

In this section, the various functions of the Point Marketplace will be outlined.

4.1 Home Page

Upon entering, the user is greeted by a welcome messenger, a list of current prices of points in demand from both the Banks and individual listings.

4.1.1 Create Account

Clicking the ‘Create Account’ button brings the user to a create account page. Here, the new user fills in the relevant information, such as their email, phone, name, password, and location, and then they click the create account. Doing so will bring the user back to the home page.

4.1.2 Login

Clicking the ‘Login’ button brings the user to the login page. Here they enter in their email and password and click login. After doing so, the user is brought back to the home page.

4.2 My Account

From any menu the user is on, (bar the Login and Create Account), clicking the ‘My Account’ button will bring the user to their account. Here, the user can view their account information, such as their name associated with the account, their email, phone number, user ID, and location. They can also view their preferred bank, and highlighted offers. Additionally, they have the option to edit their account, go to their settings, and log out.

4.2.1 Edit Account

Clicking the ‘Edit Account’ button will allow the user to edit their account information. For example, if the user wanted to change their preferred bank, they would click the ‘Edit Account’, and change it there, then save their changes.

4.2.2 Settings

After clicking the ‘Settings’ button, the user would be brought to a settings page, where they can adjust their settings. For example, if the user wanted to change their notifications, they would do so here.

4.2.3 Logout

Clicking the ‘Logout’ button will log the user out of their account.

4.2.4 Preferred Bank

Here the user can view their preferred bank. For example, a preferred bank could be; CIBC, TD, BMO, Scotiabank and/or RBC.

4.2.5 Highlighted Offers

Here the user can see current offers being provided that could be ideal to the user.

4.3 Wallet

After clicking the ‘Wallet’ button on any screen, (besides the Login and Create Account page), the user will be brought to their wallet. In their wallet, it contains the information of their connected loyalty programs, such as their points. Here they can also view the current pricing for the points out of three options; Market Value, Quick sale Value, and Merchant Value. Furthermore, the user can simply quickly add other loyalty programs that they have not added yet.

4.4 About

The about page is a page full of all the information and details you could require. It talks about how our program works and gives a summary of our project and the goal that we set out to accomplish.

4.5 Buy

The buy page is where you can purchase your points from the marketplace. Using an API, users can request to buy points from the marketplace. The API then receives the request and determines if there are enough points for purchase and creates a transaction to purchase points. The user is then given the prompt of a successful purchase.

4.6 Sell

The sell page is the page where users go to sell their points for money. Users will click the “sell points” button and then receive a prompt to select which points they wish to sell and how much. The API is then contacted and checks to see if the amount that the user wants to sell isn’t more than what the user has. If everything is okay, the user will receive money for their points.

5. Troubleshooting & Support

Common issues and their solutions can be found in this section of the user manual.

5.1 Error Messages and Behaviors

Vue.js 3 or Node.js Cannot be found

In order for you to run this program, you are required to install Node.js and Vue.js 3. If you have not successfully installed at least one of these programs then the corresponding error message will occur. To download and install node.js: **www.nodejs.org/en/download/**. To download and install vue.js 3, please follow their installation guide: **www.v3.vuejs.org/guide/installation.html**.

Dependency Errors

If any dependencies within the program provide errors or cannot function within the program, first check to ensure the dependency was properly imported. If it is locally imported, check the specific javascript code. If it was globally imported, ensure that main.js and/or router/index.js have been correctly and successfully imported. If the problem persists, re-install dependencies, make sure to check for any updates if needed.

Node.js or System Failures

If at any point an error should arise regarding Node.js or the system, the user should check their Node.js version and their Java version. In order to check for the Node.js version, type **npm -v** in the terminal. In order to check for the Java version, type **java -version** in the terminal. If you find the versions to be out of date, you should update them. This program uses the new vue.js 3 framework, given that it is in its early stages of development, you should be aware that it may need updates from time to time. However, this is not mandatory unless an upgrade/improvement of the software is required and that installation results in a system or computational failure. In order to check the view.js version, type **vue –version** in the terminal.

Firebase Failure

Firebase is a cloud service, so the first thing is to understand whether the error stems from the network or from the dependency "missing". If the error specificies network issue, check to ensure you are connected to the correct web api. You can find this in the main.js file under firebaseConfig. Compare the firebase config with the firebase web interface config, ensure the details match. If the configs are the same and the problem still persists, check the network status of the cloud server on firebase, this is found under hosting in the project directory. If the network is down or running slow, you can switch the preferred network location to another. If the error says "firebase is missing" then the dependency was either not installed correctly. To install firebase type **npm install firebase** in the terminal.

Localhost Already In Use

This error means that your default local host is already in use by another software or program. You can either terminate the current software using your default local host (most likely, localhost:8080 or localhost:3000), or you can change port options in package.json : scripts.serve=vue-cli-service serve –port 8080. You can also just serve the port on terminal run: **npm run serve – –port 3000**.

5.2 Special Considerations

Node.js

This code prototype needs Node.js to be installed on the system for the website to work. To download and install node.js: **www.nodejs.org/en/download/**.

Vue.js 3.x

The website uses the vue.js 3 framework, so ensure it is downloaded and installed. Without the framework the website will not know how to communicate properly within the code.To download and install vue.js 3, please follow their installation guide: **www.v3.vuejs.org/guide/installation.html**.

Firebase

Firebase is used for the backend of the web application. It handles user authentication and the database. Without it, a user would not be able to get past the authorization to access the protecting routes(pages). To install firebase type **npm install firebase** in the terminal. If you would like to make your own local authentication you can alter the firebaseconfig in the main.js file.

Dependency Installs

Numerous dependencies are installed within the website. These dependencies are crucial to the website's execution. So do not remove or alter any of the dependencies unless you have an alternative method or are updating them. However if either occurs you will need to fix the code following it.

Internet Connection

A stable internet connection is required if you wish to use the website and get past the account authentication. Without the internet you will remain on the unprotected views(pages), missing the actual function of the site.

6. Product Documentation

This final prototype was built using the Vue.js 3 framework which is a powerful tool for building web applications. It allows for dynamic one-page web applications. Along with vue.js it uses react routers to route views(pages). This is matched with a firebase to protect certain directories from unauthenticated unders. Firebase handles the bankend cloud information, user authentication and database. With firebase we can tailor the information seen by the user using freebase firestore which is a real-time cloud storage solution.

The reason behind using Vue.js for the web interface was due to the technical knowledge of the tool within the team. It is also just a great and powerful tool for making websites. It’s capabilities are unmatched within the industry. However, Vue.js 3.0 might have been the wrong idea given how new it is. Due to its little development, poor reliability and minute connectivity with other tools, vue.js 2 would have been a better idea. Although given that the website was made in version three, within time this will prove to be beneficial.

We decided to use firebase because of it’s low cost and simplistic api. It served our application very well and the integration between the database and authentication was considerably easy. Although, there was not much support regarding the implementation of firebase on a vue.js 3 application, given it’s little development. Yet, given passion and perseverance it was successful in providing what we needed. A better alternative would have been microsoft azure, but we did not have the time nor resources to implement a bank end system with azure.

Hosting became a big issue in the end. When it came to compiling the website and deploying it on a server we just didn’t have the time to execute it. Deploying it locally worked fine and hosting it on a personal machine worked but given there was no security for the personal system we could not sustain this decision for public use. However, it does in fact work. It would be recommended to create a secure linux server to run the website as security is extremely important.

Within the infrastructure of the code, recursion of the authentication status of the user was a concern of ours. This was something we had predicted before starting the final project. The question was how were we going to limit our user reads so we were not calling the server every second or so. This would be an issue as it causes unnecessary traffic on the cloud server. After research it was realized that Firebase has an implementation that allows for the user state to only read if the user were to make a change to their authentication state. Using firebases api, the code looked like this:

```
firebase.auth().onAuthStateChanged(() => {
```

This was crucial for our minimal traffic to the cloud service. Even with hours of testing the authentication and reading user data, it only hit 1,300 Reads.

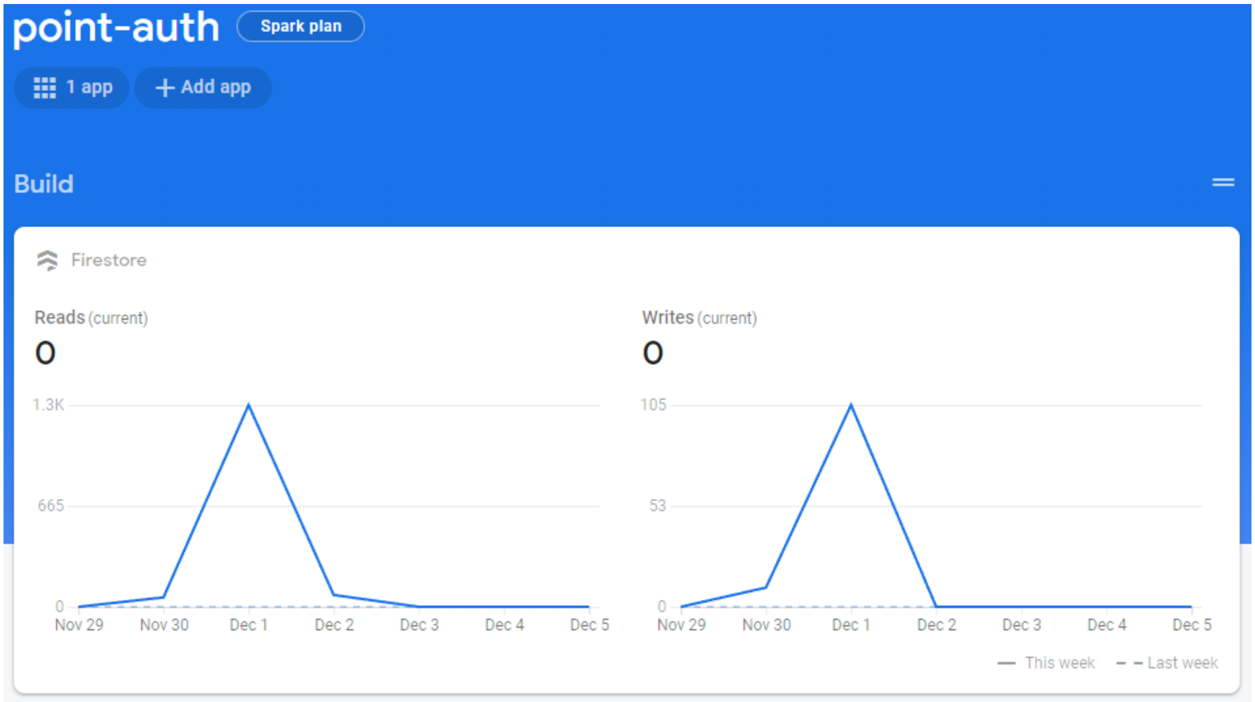


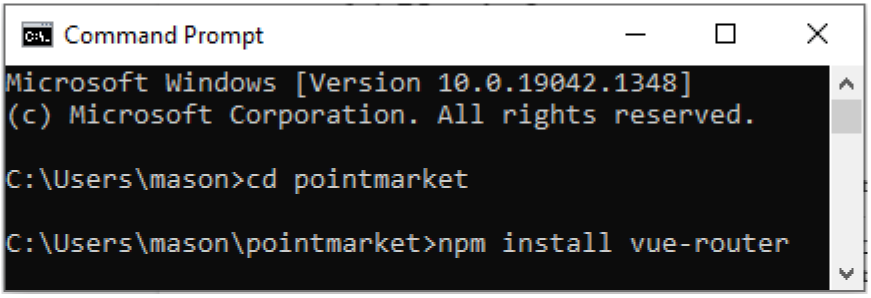
Figure 5: Firebase Reads/writes analytics

Ultimately this could have been done even better if a custom api was made, but given the time that was not possible.

6.1 Vue.js 3

6.1.1 BOM (Bill of Materials)

- **Node.js:** A backend javascript environment, allows javascript environments to run outside of a browser. It is a necessity to run Vue.js and any other tools like it. It is free and fairly easy to install. To download and install node.js: **www.nodejs.org/en/download/**.
- **Vue.js 3:** A free and open source web framework for building web applications or more specifically web interfaces. It is the foundation of the point marketplace. To download and install vue.js 3, please follow their installation guide: **www.v3.vuejs.org/guide/installation.html**.
    - **Vue-Router:** Is a required dependency for this vue application as it is what handles the routes(directory). To install Vue-Router, type **vue install vue-router** in the terminal in the project directory.

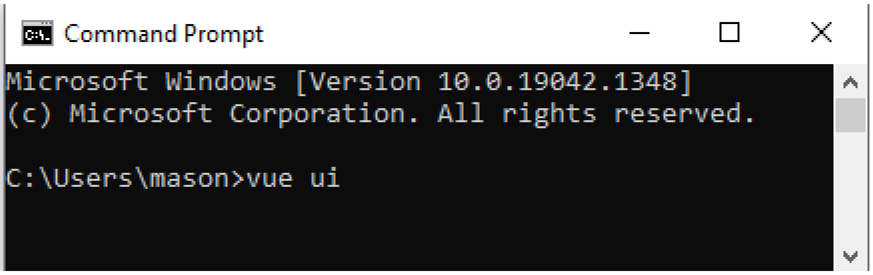


```
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mason>cd pointmarket

C:\Users\mason\pointmarket>npm install vue-router
```

Figure 6: Command Prompt vue router install Example

6.1.2 Equipment List

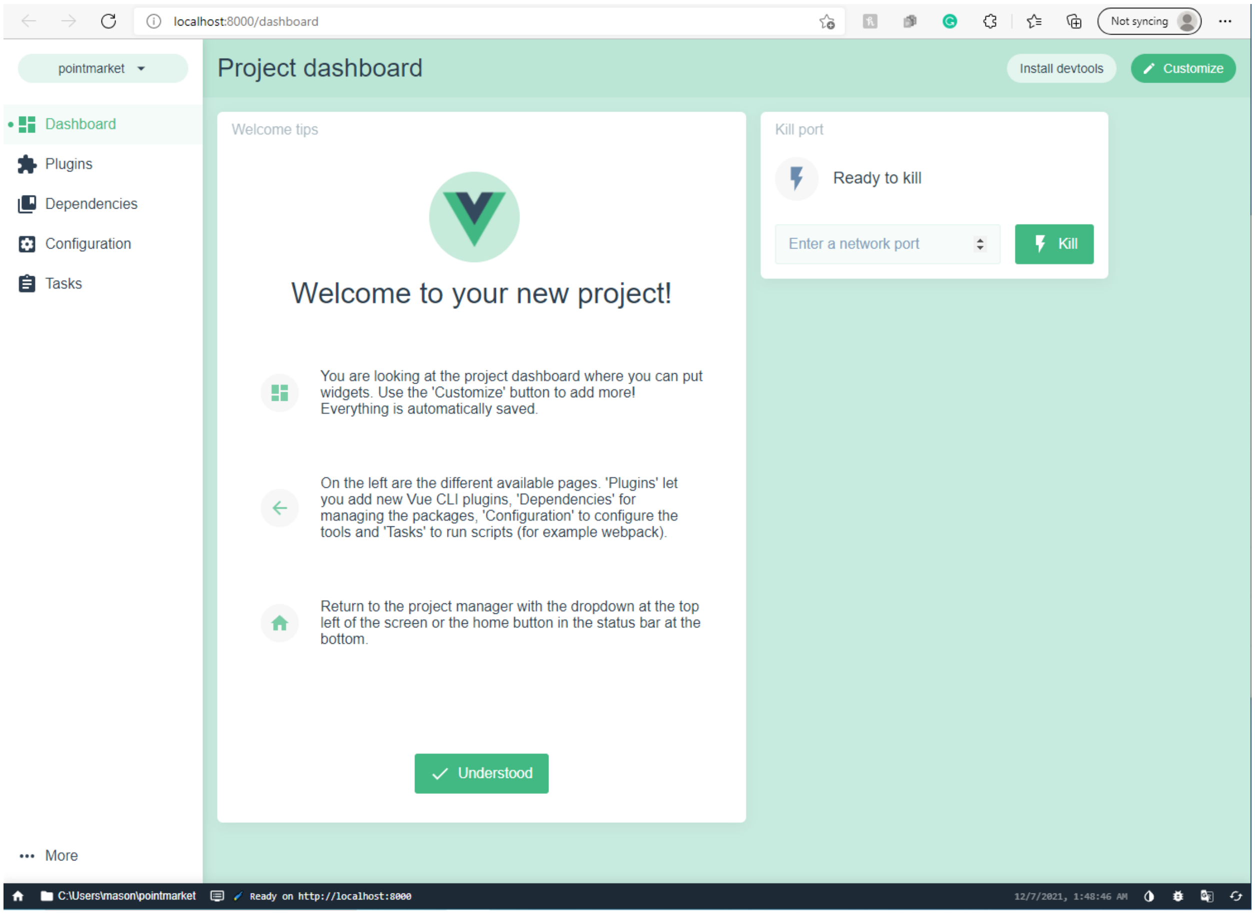
- After you install vue.js 3, the vue ui can be accessed by typing **vue ui** in the terminal.



```
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mason>vue ui
```

Figure 7: Command Prompt vue ui open example

After the execution of this command a new window will open in localhost. It should look something like this:



Welcome to your new project!

You are looking at the project dashboard where you can put widgets. Use the 'Customize' button to add more! Everything is automatically saved.

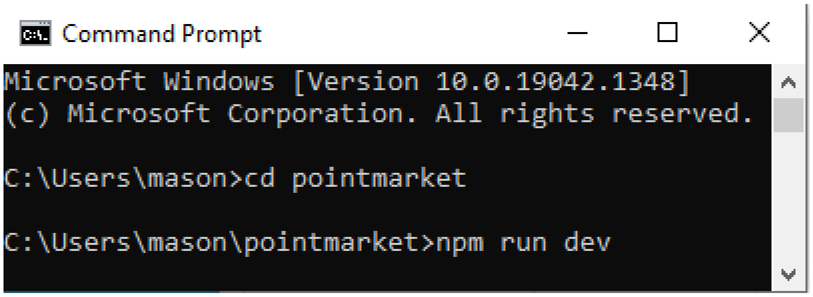
On the left are the different available pages. 'Plugins' let you add new Vue CLI plugins, 'Dependencies' for managing the packages, 'Configuration' to configure the tools and 'Tasks' to run scripts (for example webpack).

Return to the project manager with the dropdown at the top left of the screen or the home button in the status bar at the bottom.

Figure 8: Vue ui Dashboard

Using this UI you can add or remove plugins and dependencies. You can also check current versions of libraries in extensions, alter configuration and serve localhost. Think of it as a visual terminal.

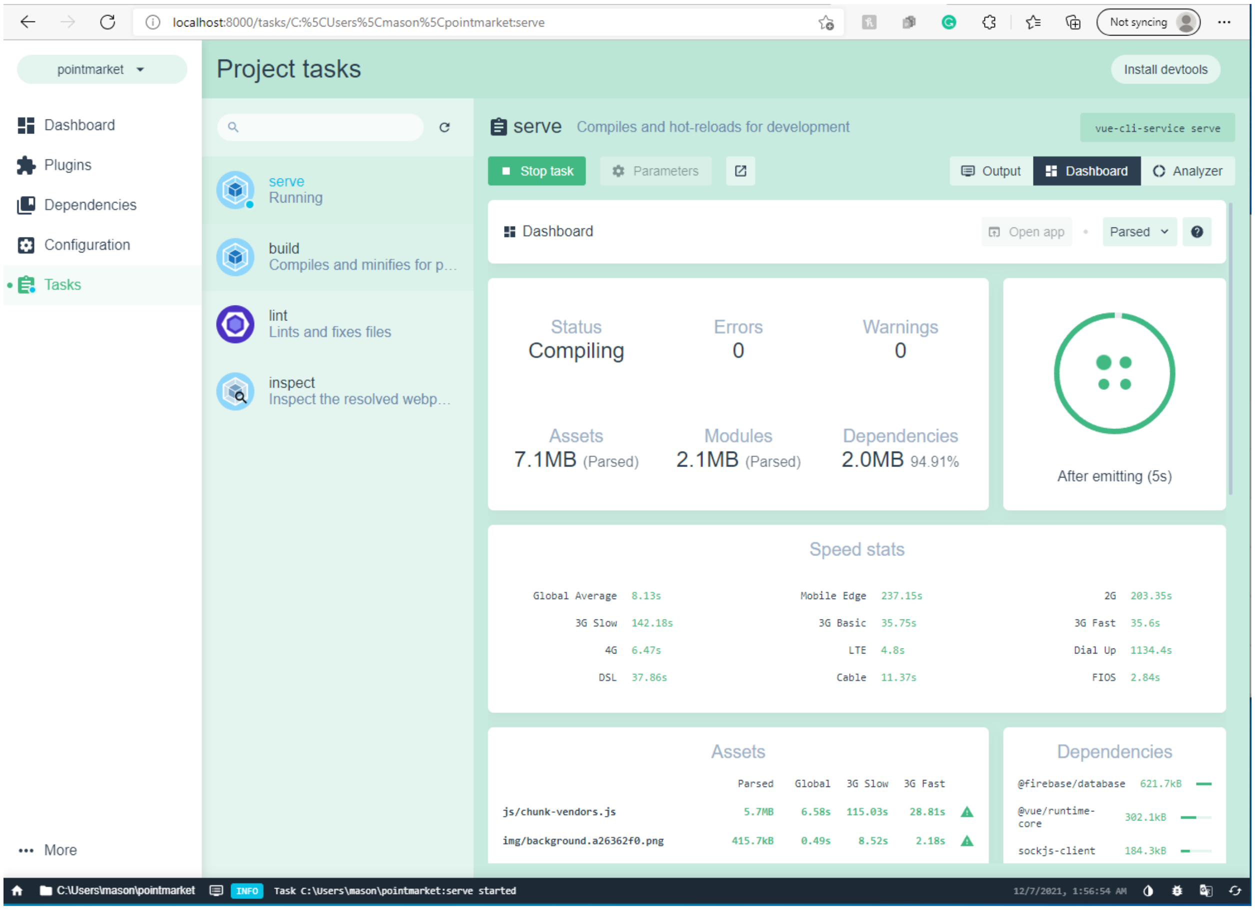
- You can serve(run) the website in one of two ways. (1) Using the terminal:



```
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mason>cd pointmarket

C:\Users\mason\pointmarket>npm run dev
```

Figure 9: Command Prompt run website through npm

(2) Using the vue ui: Go to”Tasks”, click serve and run task



Status	Errors	Warnings
Compiling	0	0

Assets	Modules	Dependencies
7.1MB (Parsed)	2.1MB (Parsed)	2.0MB 94.91%

Speed stats

Global Average	8.13s	Mobile Edge	237.15s	2G	203.35s
3G Slow	142.18s	3G Basic	35.75s	3G Fast	35.6s
4G	6.47s	LTE	4.8s	Dial Up	1134.4s
DSL	37.86s	Cable	11.37s	FIOS	2.84s

Assets

	Parsed	Global	3G Slow	3G Fast
js/chunk-vendors.js	5.7MB	6.58s	115.03s	28.81s
img/background.a26362f0.png	415.7kB	0.49s	8.52s	2.18s

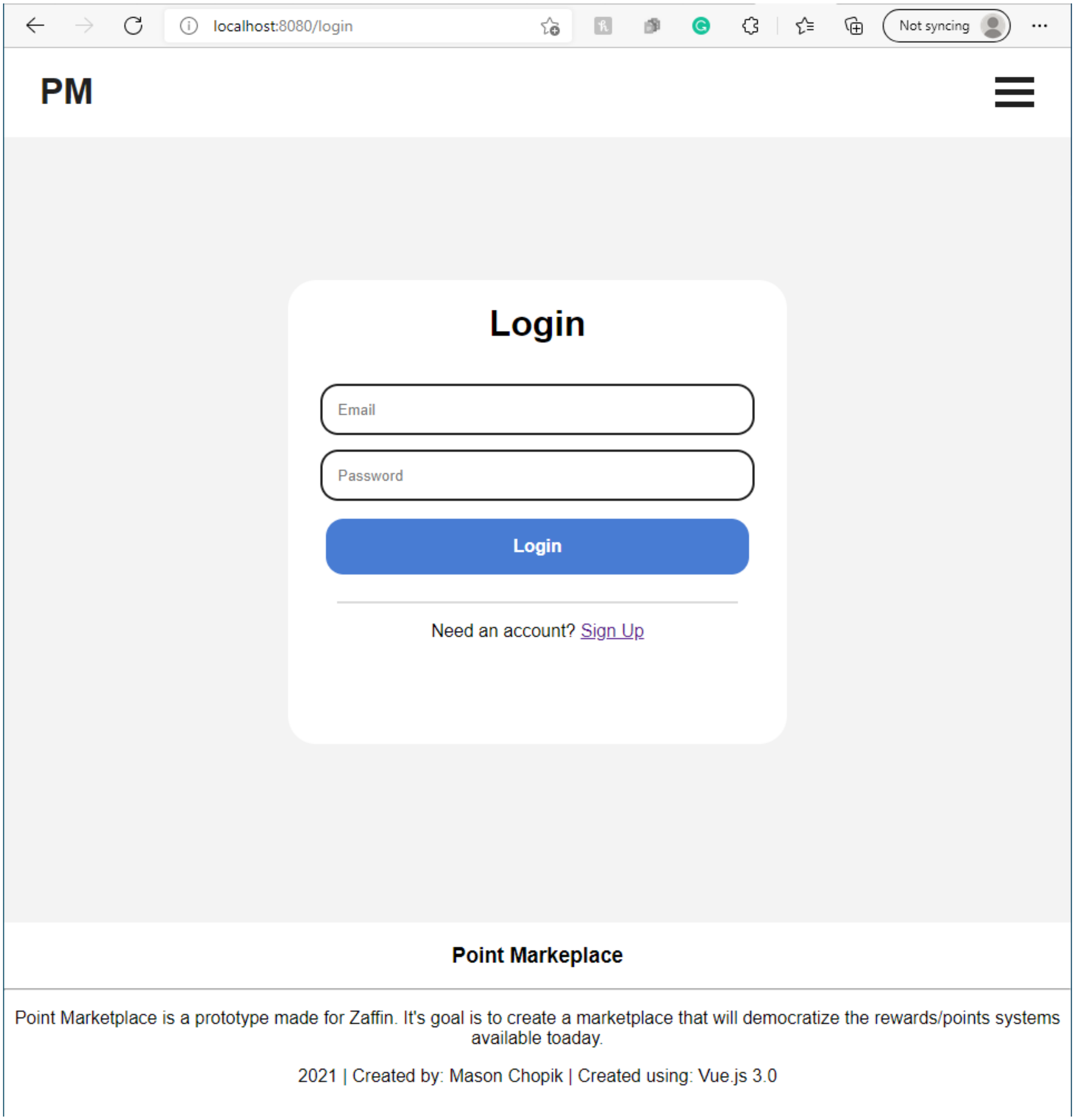
Dependencies

@firebase/database	621.7kB
@vue/runtime-core	302.1kB
sockjs-client	184.3kB

```
C:\Users\mason\pointmarket  INFO  Task C:\Users\mason\pointmarket:serve started  12/7/2021, 1:56:54 AM
```

Figure 10: Serving(running) website through vue ui

After the website is done compiling it will open in a localhost port (most likely 8080 or 3000).



Need an account? Sign Up

Point Marketplace is a prototype made for Zaffin. It's goal is to create a marketplace that will democratize the rewards/points systems available toaday.

2021 | Created by: Mason Chopik | Created using: Vue.js 3.0

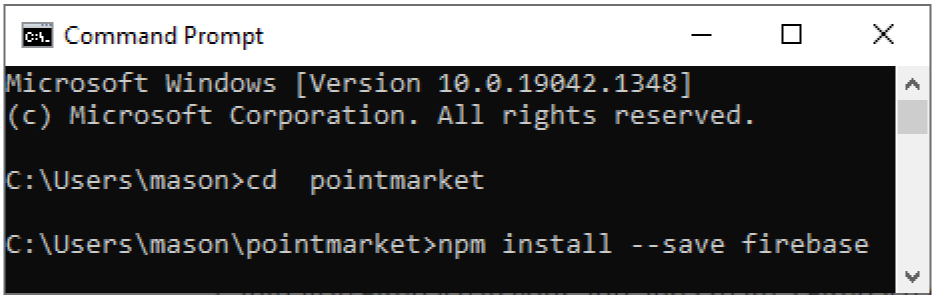
Figure 11: Login Page

Now you can interact and use the application.

6.2 User Authentication

6.2.1 BOM (Bill of Materials)

- **Reliable Internet Connection:** In order to connect to a cloud service you need a reliable internet connection. Without this connection, a user will not be able to authenticate or make an account to pass the protected routes(pages).
- **Firebase:** Is a cloud service platform developed by Google. It has numerous products, but in this case, only firebase authentication and firestore were used. Firebase authentication dealt with the user authentication, once the user was authenticated it connected with the firestore database to use user data. Firebase is a pay-as-you-go, however, they are extremely generous, so as long as traffic and server use are reasonably low it will be free. In order to install firebase:



```
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mason>cd  pointmarket

C:\Users\mason\pointmarket>npm install --save firebase
```

Figure 12: Command Prompt firebase install example

6.2.2 Equipment List

- Firebase config is where the code points to the correct server. The firebase config can be in a separate file or in our case in the main.js file:

```
const firebaseConfig = {

    apiKey: "AIzaSyDf2g3GEW5xGP9mtU_5RG6giMPt49ETN-0",

    authDomain: "point-auth.firebaseapp.com",

    projectId: "point-auth",

    storageBucket: "point-auth.appspot.com",

    messagingSenderId: "324401325825",

    appId: "1:324401325825:web:867ba2af81a945d1fcdd00"

};
```

If you prefered to run on your own server you could create your own project on firebase and use your own custom firebaseConfig values.

* Firebase needs to be imported to do so add the following; `import firebase from 'firebase';`*

- Following the definity of the firebaseConifg, it needs to be initialized in the app:

```
firebase.initializeApp(firebaseConfig);
```

- Now that firebase is initialized within the app it's api and cloud services can be used.

- In order to protect unAuthorized users from accessing views(pages) they should not have access to, the following code was used:

```
  setup() {
    const router = useRouter();
    const route = useRoute();

    onBeforeMount(() => {
      firebase.auth().onAuthStateChanged((user) => {
        if (!user) {
          router.replace('/login');
        } else if ((user) && (route.path == "/login" || route.path
== "/register")) {
          router.replace('/wallet');
        }
      })
    });
  }
```

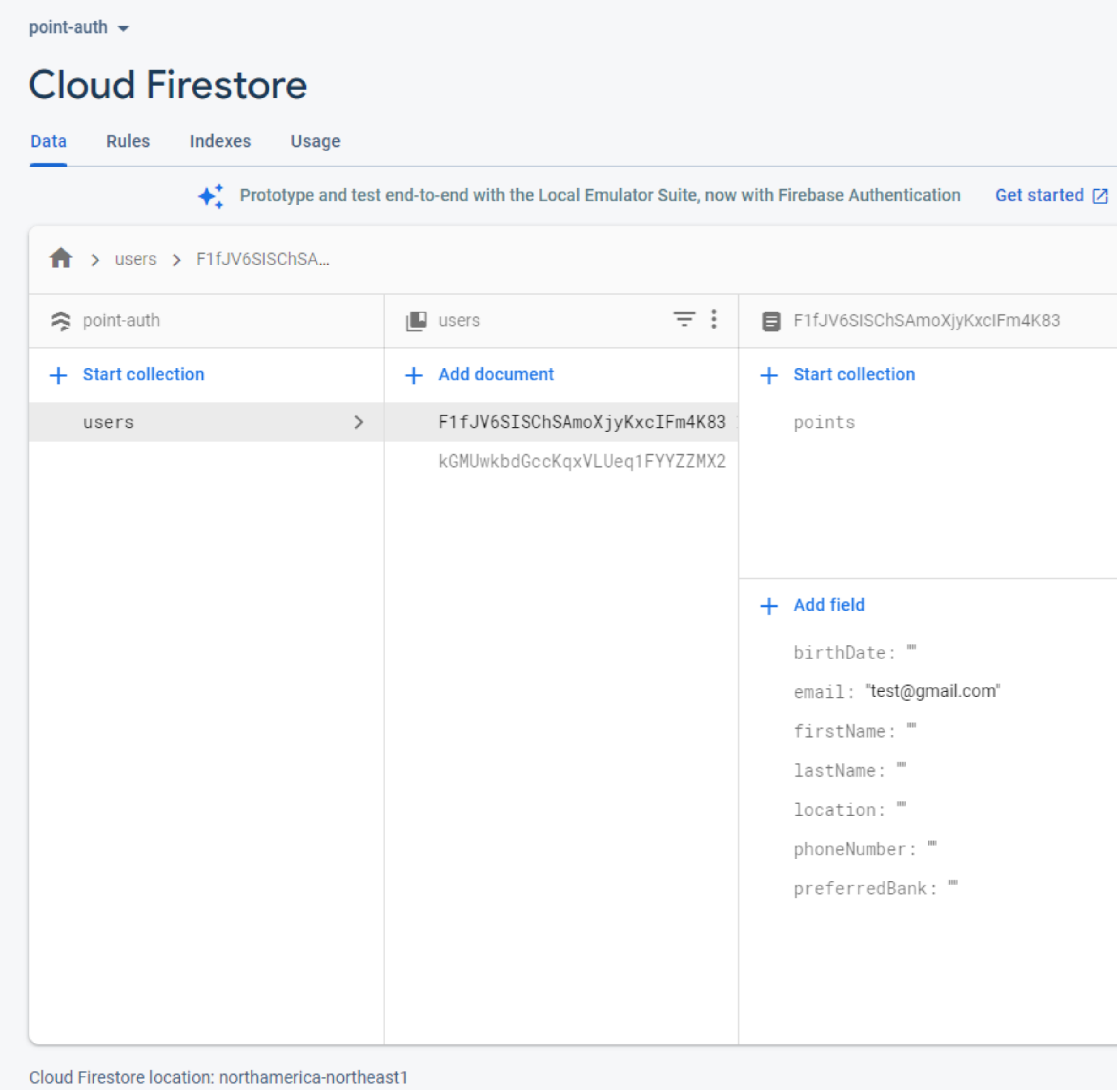
Simply put, whenever the user makes a change to their authentication state, it checks to see if they are a user. If they are not a user it will transfer their directory to the login page. If they login and their authorization status is changed to a valid user then it will direct them to the wallet page. Without this instance any user could access any part of the website, even if they did not have a valid account.

- Whenever a new user is created a data structure is made within firestore, this is executed by the store/index.js file which is called by:

```
created() {
    firebase.auth().onAuthStateChanged((user) => {
      this.store.commit("updateUser", user);
      if(user) {
        this.store.dispatch("getCurrentUser");
      }
    })
  },
```

The store file contains actions, mutations, state and modules. Theses handle what to do with the firestore data.

- When a user is first created, they are given a unique UID which connects to the cloud firestore.


point-auth
Cloud Firestore
Data
Rules
Indexes
Usage
Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication
Get started
users > F1fJV6SISChSA...
point-auth
users
F1fJV6SISChSAmoXjyKxcIFm4K83
Start collection
Add document
Start collection
Add field
Cloud Firestore location: northamerica-northeast1

users

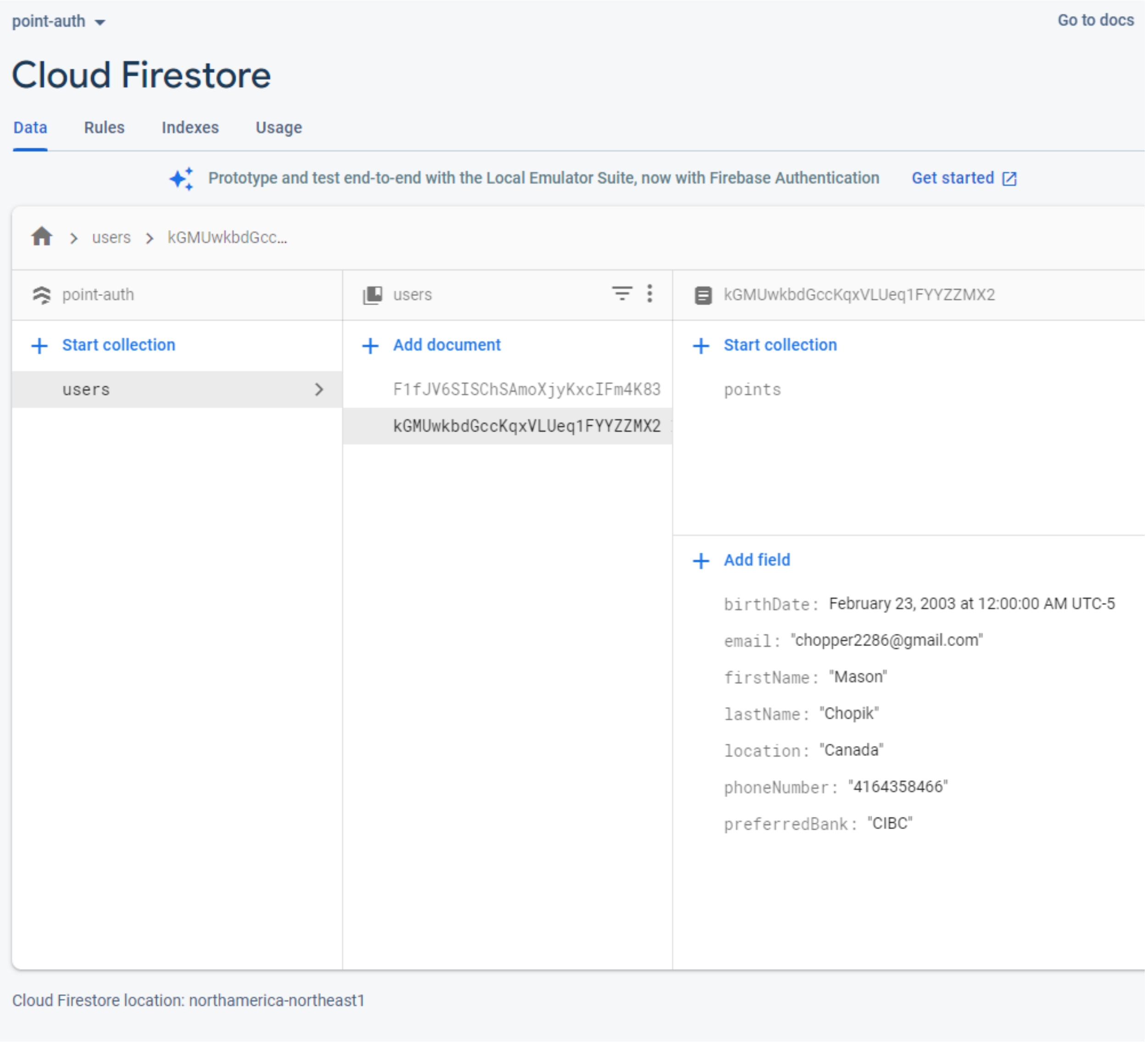
F1fJV6SISChSAmoXjyKxcIFm4K83
kGMUwkbdGccKqxVLUeq1FYYZZMX2

points

birthDate: ""
email: "test@gmail.com"
firstName: ""
lastName: ""
location: ""
phoneNumber: ""
preferredBank: ""

Figure 13: Cloud Firestore New User Data

Initially their data is empty, however on the user profile page the user can populate their data:



point-auth

Go to docs

Cloud Firestore

Data Rules Indexes Usage

Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication Get started

users > kGMUwkbdGcc...

point-auth

+ Start collection

users

users

+ Add document

F1fJV6SISChSAmoXjyKxcIFm4K83

kGMUwkbdGccKqxVLUeq1FYYZZMX2

kGMUwkbdGccKqxVLUeq1FYYZZMX2

+ Start collection

points

+ Add field

birthDate: February 23, 2003 at 12:00:00 AM UTC-5

email: "chopper2286@gmail.com"

firstName: "Mason"

lastName: "Chopik"

location: "Canada"

phoneNumber: "4164358466"

preferredBank: "CIBC"

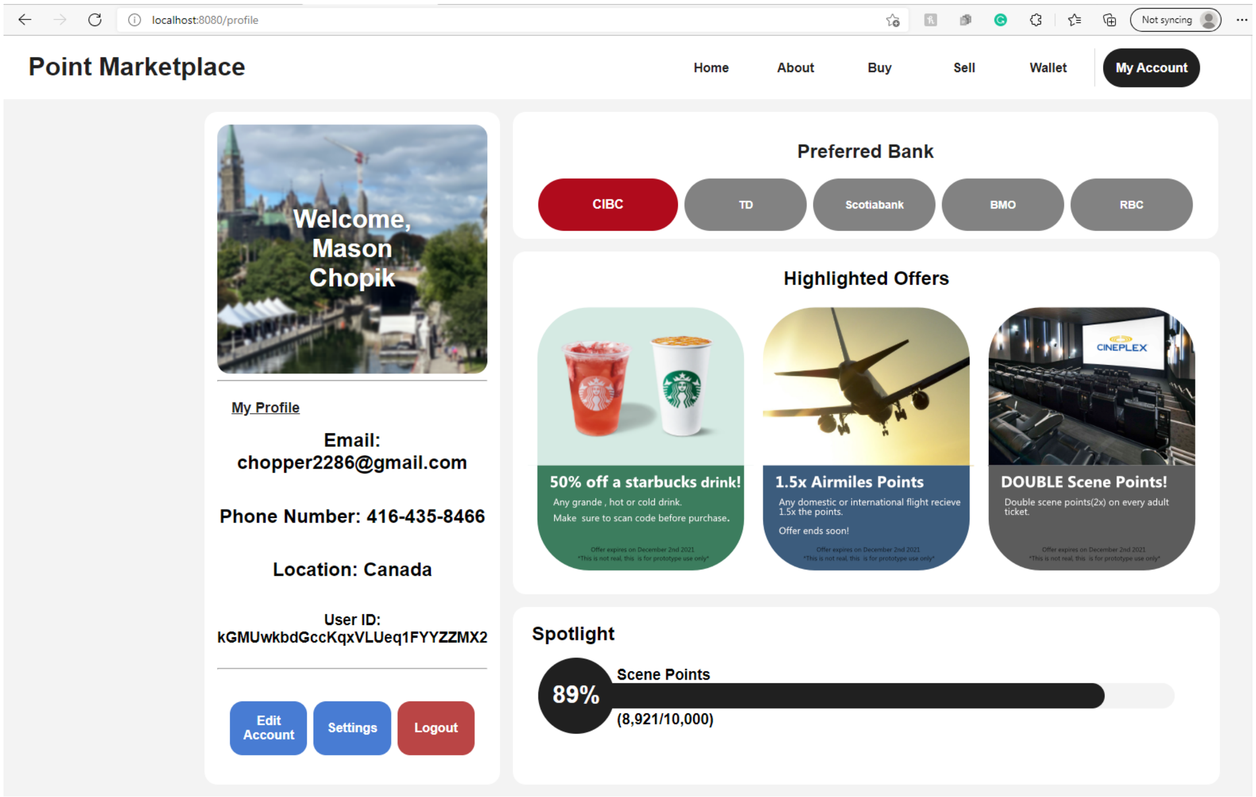
Cloud Firestore location: northamerica-northeast1

Figure 14: Cloud Firestore Populated User Data

6.3 Testing & Validation

Throughout the creation of the final prototype we tested the capabilities and feasibility of our design. The reason we went for the high fidelity design was to ensure our design was feasible and proper. So during the production we were testing the code to ensure it both worked and was actually realistic and possible. In addition, the high fidelity allowed for limitless customization and the possibility for it to turn into an actual product.

We also tested the user interface and user experience with other classmates and people within our networks. This testing actually resulted in the customizable account hub:



Email: chopper2286@gmail.com

Phone Number: 416-435-8466

Location: Canada

User ID: kGMUwkbdGccKqxVLUeq1FYYZZMX2

50% off a starbucks drink!
Any grande , hot or cold drink.
Make sure to scan code before purchase.
Offer expires on December 2nd 2021
This is not real, this is for prototype use only

1.5x Airmiles Points
Any domestic or international flight recieve 1.5x the points.
Offer ends soon!
Offer expires on December 2nd 2021
This is not real, this is for prototype use only

DOUBLE Scene Points!
Double scene points(2x) on every adult ticket.
Offer expires on December 2nd 2021
This is not real, this is for prototype use only

Scene Points (8,921/10,000)

Figure 1: Profile Page

This was prompted by many of our test users, as they wanted something where everything was displayed to them at once.

Another ongoing test throughout the project was testing the cloud server and user authentication. There were numerous issues and tests we needed to take in order to get where it is now but the biggest one was ensuring the user was authenticated when switching pages. Users should not be validated on every page request, that would defeat the purpose of the one-page application and it would stress the cloud server so through testing and research we found an alternative. Using the firebase api we were only able to check if the user changed their user authentication state. This resulted in a reasonable amount of cloud reads, instead of loading the server with requests. This is shown below.

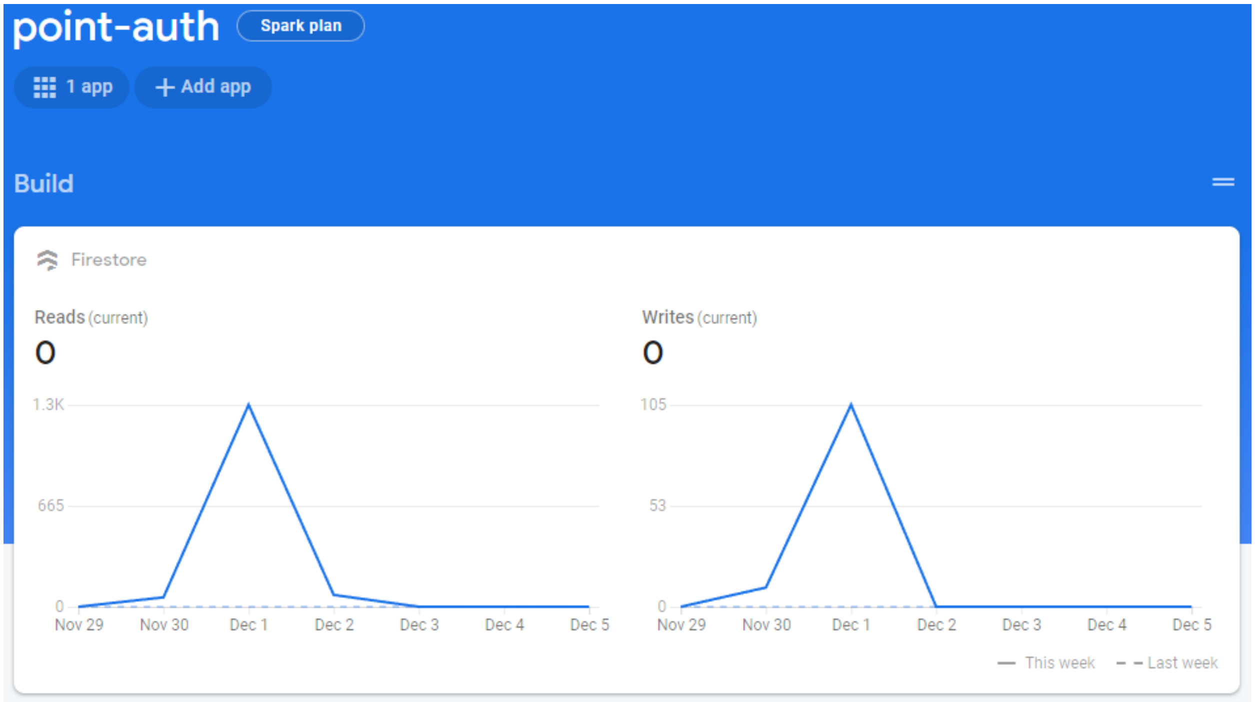


Figure 5: Firebase Reads/writes analytics

7. Conclusions and Recommendations for Future Work

In conclusion, our point marketplace is a user-friendly loyalty program platform that democratizes the use of points using the free-market approach. The application gives users the freedom to choose how they want to use their points by selling the unwanted points to the bank and buying the points they want.

The user manual provides simple instructions to guide the user on using the platform and gives a better understanding of how to use it for their best benefits.

While conducting this project, we learned many lessons, some of the important lessons we learned:

- We learned the importance of communication while working in a team. We learned how to make online communication effective. For every deliverable we communicated using Discord mostly and Zoom during the lab time, we also had in-person meetings in the Learning Crossroads (CRX). We discussed the tasks assigned to each member based on their strength and the deadline for each task. And after assigning the tasks we also used Discord to ask questions and also ask for help from other team members if needed. We let each other know if something urgent came up and how we will handle it. Overall, we had effective communication by constantly communicating with each other and this helped our team to achieve our final product.

- We learned how to plan a project while also working in a team. First, we learned how to manage our time in order to finish the deliverable on time by setting deadlines for ourselves and we also used wrike and Discord to help keep track of that for example we had a channel on Discord where everyone can let the other team members that they have finished their part of the deliverable or their task and that helped because it let the other team member know that they can start working on their part if it is related to that part. Also Wrike helped by assigning the tasks and the deadline since it sends reminder notifications. Since we followed this plan, our team was able to complete all the deliverables on time.

- Since the project was a software-based project we had to learn coding and since some of the group members are familiar with Vue.js 3 application Firebase Authentication and Firestore which is what we used for our platform.

- One of the most important lessons was the design process. We learned how to apply design thinking in solving real world problems. We learned how having a design process allows us to come up with the best solution in very specific steps and avoid missing important aspects. We followed the steps of design thinking and iterated between them based on the feedback we received from the client from benchmarking and empathizing to achieve the best solution. It also allows us as a team to combine all our ideas and come up with the best one because after coming up with a design criteria,we generated our individual designs first and then we came up with the best one to work on and make prototypes and test them. It also showed us how important the testing stage was because it allowed us to improve our prototype every time it fails and finally have a functional product. Overall, having a design process helped us stay organized and helped us focus on all important aspects of the design.

- Since the project included three presentations, we also learned presentation skills and how to apply them like how to make our presentation engaging by speaking loudly and using body language and also work on the visuals of our presentations (no long texts but rather informative visuals).

- Lastly, we also learned how to work with a real client and solve a real world problem and that was a new experience that we all enjoyed but also felt more responsible and professional.

For our future work, if our team had more time to work on this project, we would have the website running so that users can use it and create accounts rather than just a demo. Additionally, in order to make the stock market work, we would have a long-term plan. Also, we would like to make an app instead of a website which makes it easier for users.

APPENDICES

8. APPENDIX I: Design Files

Table 3: Referenced Documents

Document Name	Document Location and/or URL	Issuance Date
Team Contract and Project Management	https://uottawa.brightspace.com/d2l/common/viewFile.d2lfile/Database/NTc0Njg3Ng/Team%201%20Project%20Contract.pdf?ou=241561	September 23, 2021
Needs Identification and Problem Statement	https://uottawa.brightspace.com/d2l/common/viewFile.d2lfile/Database/ODY4NTUzOQ/Deliverable%20B%20-%20Needs%20Identification.pdf?ou=241561	October 3, 2021
Design Criteria	https://uottawa.brightspace.com/d2l/common/viewFile.d2lfile/Database/ODc1MzM3OQ/Deliverable%20C%20-%20Design%20Criteria%20(1).pdf?ou=241561	October 7, 2021
Conceptual Design	https://uottawa.brightspace.com/d2l/common/viewFile.d2lfile/Database/ODkwNTEwNw/Deliverable%20D%20-%20Technical%20Document.pdf?ou=241561	October 18, 2021
Project Plan and Cost Estimate	https://uottawa.brightspace.com/d2l/common/viewFile.d2lfile/Database/ODk5MzI5Mw/Deliverable%20E.pdf?ou=241561	October 24, 2021
Prototype I and Customer Feedback	https://uottawa.brightspace.com/d2l/common/viewFile.d2lfile/Database/OTEwNjk1OA/Deliverable%20F-%20Technical%20Document.pdf?ou=241561	November 4, 2021
Prototype II and Customer Feedback	https://uottawa.brightspace.com/d2l/common/viewFile.d2lfile/Database/OTMzODQyNQ/Deliverable%20G%20Redo.pdf?ou=241561	November 11, 2021
Prototype III and Customer Feedback	https://uottawa.brightspace.com/d2l/common/viewFile.d2lfile/Database/OTQyODg5Mg/Deliverable%20H-%20Technical%20Document.pdf?ou=241561	November 25, 2021
MakerRepo	https://makerepo.com/NBlaney/1046.point-market	N/A

9. APPENDIX II: Additional Documents

Table 4: Additional Documents

Document Name	Document Location and/or URL	Issuance Date
Original Stock Code from Prototype I	https://uottawa.brightspace.com/d2l/common/viewFile.d2lfile/Database/OTEwNjk1OQ/Stock%20Code.pdf?ou=241561	November 4, 2021