Using Spotify’s Web API to retrieve information about a user's listening habits

Joel Deighton

100130838

A report submitted to Jodrey School of Computer Science, Acadia University, for:

COMP 4983

September 21st, 2020 - December 11th, 2020

Supervisor

Jamie Symonds

December 1st, 2020

Table of Contents

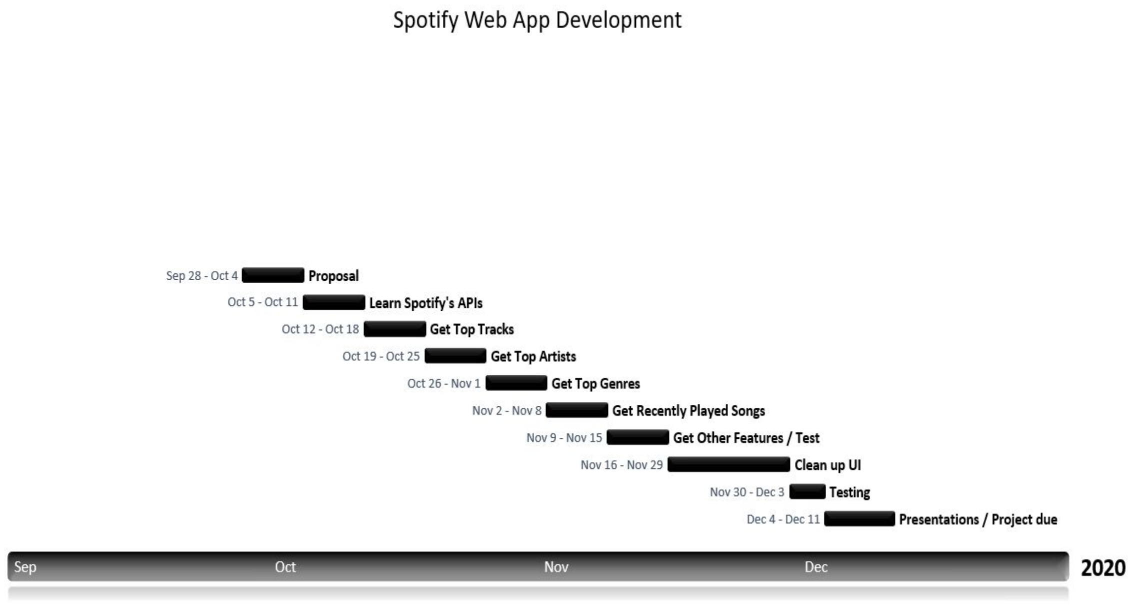
Pre-Development

I want to create a web application relying on Spotify's APIs that can produce lists of:

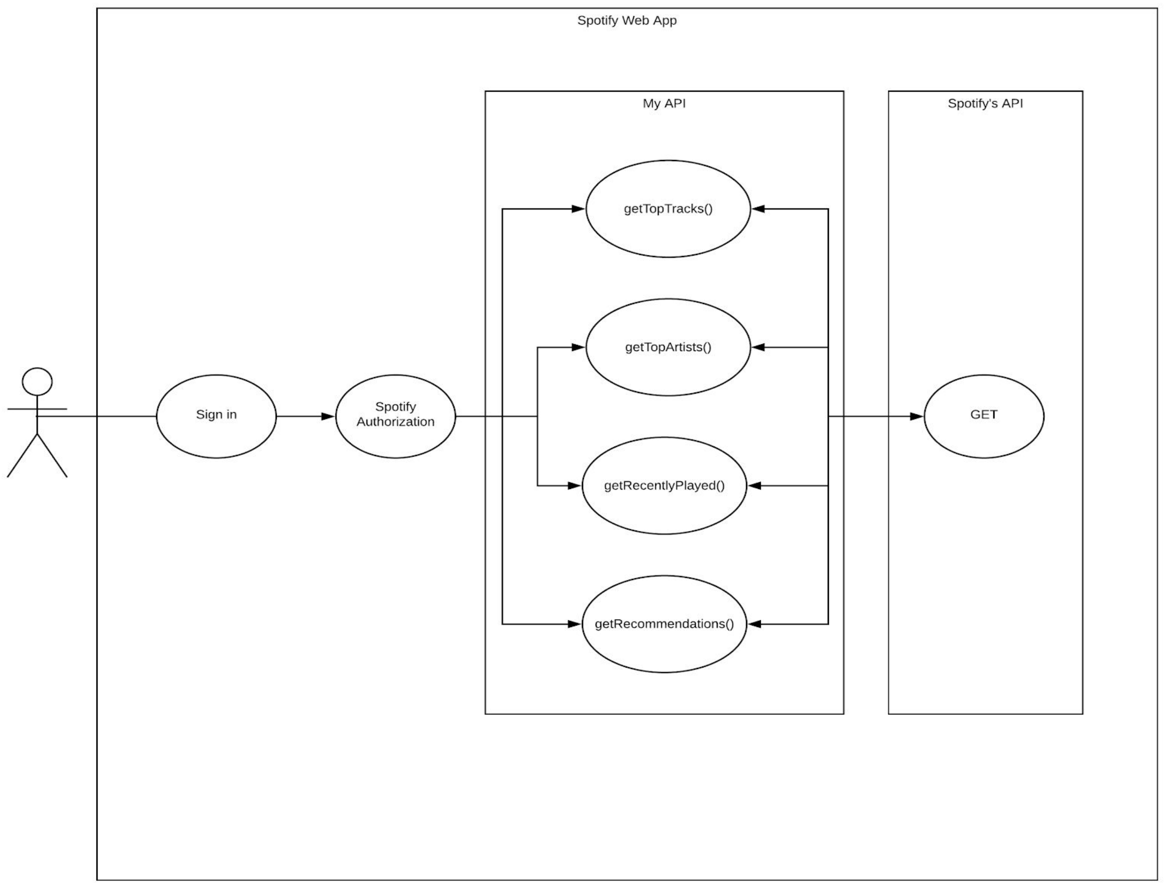
- Your top tracks
- Your top artists
- Your recently played songs
- Song recommendations based off of your top artists

Once I have implemented the basic features written above, I will be looking at other features to implement. An example of this would be generating playlists based off of the song recommendations feature. This application has the potential to be scaled and deployed to the cloud. For the development of the web app will be using Vue.js for the first time, as well as HTML and CSS. My development timeline, use case diagram, and user interface mockup are shown on the next couple of pages. Not all testing days will be used - leftover testing days will be used to make up for things I have not implemented on time.

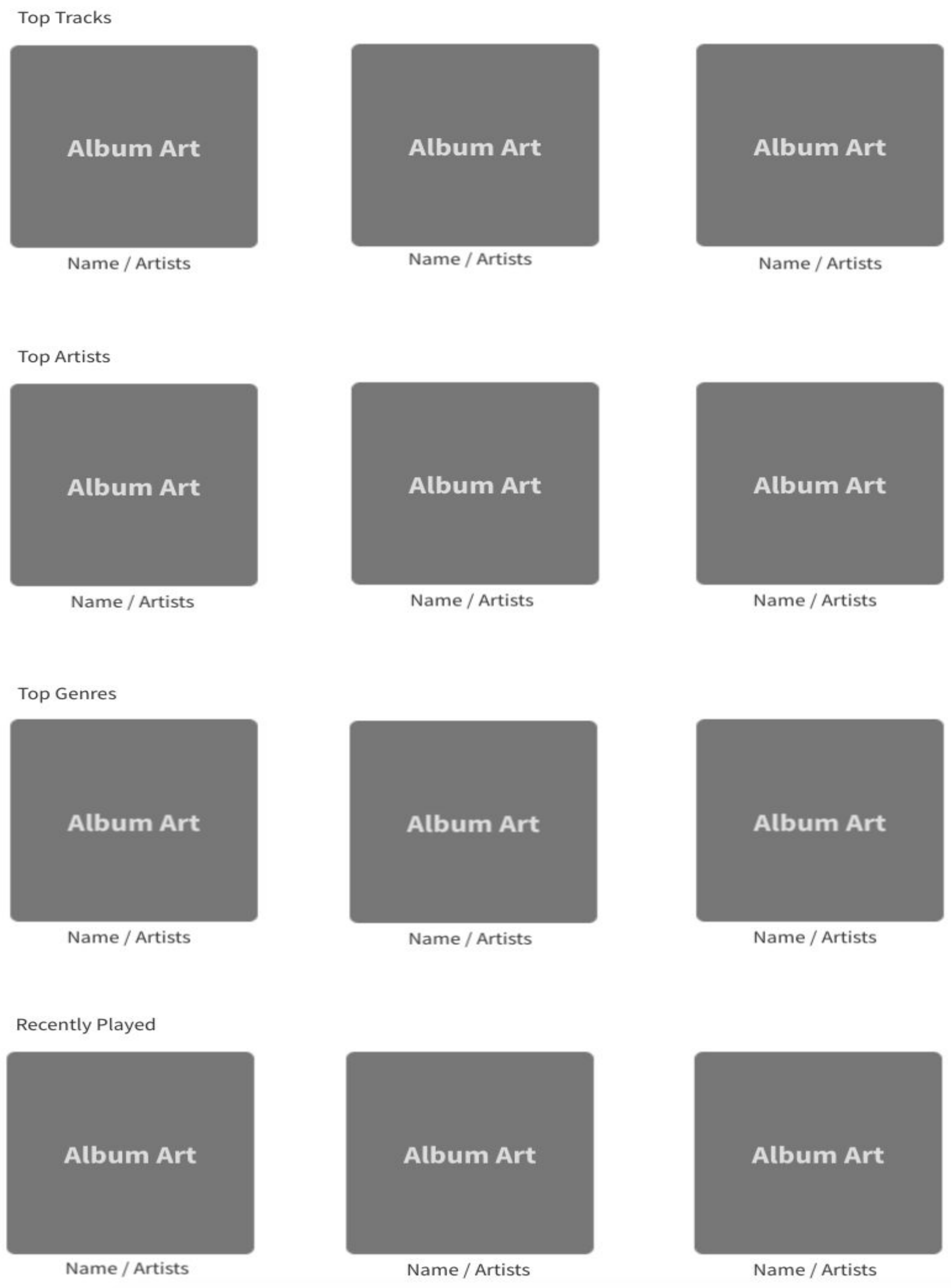
Gantt Chart



Use Case

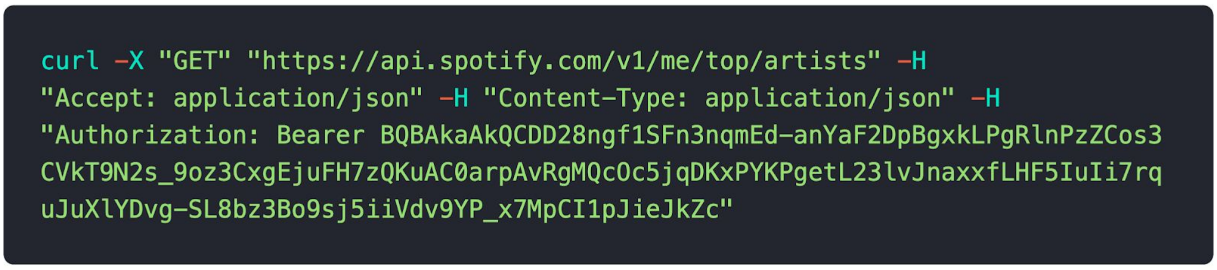


UI Mockup


Top Tracks
Album Art
Name / Artists
Album Art
Name / Artists
Album Art
Name / Artists
Top Artists
Album Art
Name / Artists
Album Art
Name / Artists
Album Art
Name / Artists
Top Genres
Album Art
Name / Artists
Album Art
Name / Artists
Album Art
Name / Artists
Recently Played
Album Art
Name / Artists
Album Art
Name / Artists
Album Art
Name / Artists

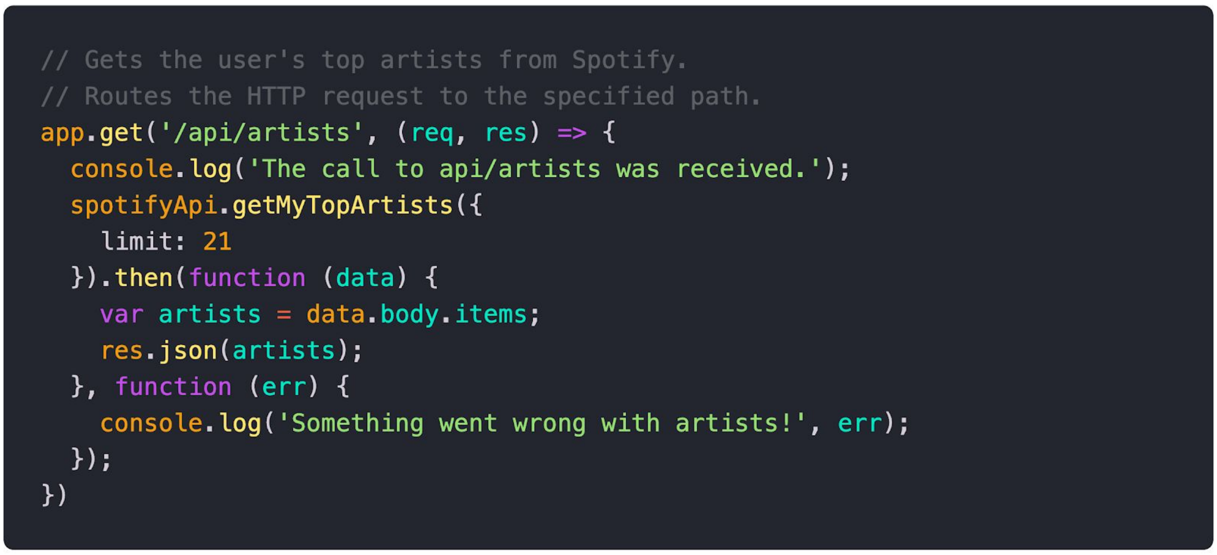
Introduction

I have spent the past three months creating a web-based application for Spotify that retrieves a user's listening habits for my capstone project. I used Vue.js, HTML and CSS for the front end development and a Node.js wrapper that was built for Spotify's API for the back end development. Wrappers are used to simplify programming by abstracting the details of the function that they are written to call, and they are useful when working with third-party applications. Since this is an application for Spotify, I had to use their API to get access to any data. I also used my API that I created to make calls to Spotify's API, which was done to prevent the exposure of any sensitive data. Here is a photo using the curl command (a command-line data transfer tool) to get a user's top artists:



```
curl -X "GET" "https://api.spotify.com/v1/me/top/artists" -H
"Accept: application/json" -H "Content-Type: application/json" -H
"Authorization: Bearer BQBAkaAkQCDD28ngf1SFn3nqmEd-anYaF2DpBgxkLPgRlnPzZCos3
CVkT9N2s_9oz3CxgEjuFH7zQKuAC0arpAvRgMQcOc5jqDKxPYKPgetL23lvJnaxxfLHF5IuIi7rq
uJuXlYDvg-SL8bz3Bo9sj5iiVdv9YP_x7MpCI1pJieJkZc"
```

Here is a photo of the wrapper function that I used to do the same thing:



```
// Gets the user's top artists from Spotify.
// Routes the HTTP request to the specified path.
app.get('/api/artists', (req, res) => {
  console.log('The call to api/artists was received.');
  spotifyApi.getMyTopArtists({
    limit: 21
  }).then(function (data) {
    var artists = data.body.items;
    res.json(artists);
  }, function (err) {
    console.log('Something went wrong with artists!', err);
  });
})
```

Body

Roughly three months ago I set a couple of goals for myself. The first was to produce a fully-deployed web app that was capable of displaying a user's favourite songs, favourite artists, and recently played songs. I also wanted it to be able to make song recommendations and generate playlists all based on your favourite artists on top of that, however, I ended up cutting out the playlist feature due to time constraints. The other goal of mine was to simply to take advantage of this opportunity to learn something new.

I did not choose this project topic because I knew how to start it or because I knew what languages and frameworks I was going to use for development, I chose it because it was just something that I was interested in and I wanted to see the development process. I started learning about Spotify's API by watching YouTube videos and reading articles about how it is used, as many students would.

This eventually led me to GitHub where I found a simple program that used Spotify's API to authorize a user and display the song they are currently listening to. Even though it was not what I wanted to do, I thought it would be helpful. I quickly realized that I was wrong and the example was useless. That was the case for the majority of examples I came across. A lot of them only authorized the user on a basic level and did not deal with one of the most difficult parts of the project, which was the authorization token.

After explaining my situation to Jamie, he sent me an article about making API calls with Vue and Express. The article walked through the basics of:

- Installing Node.js and its package manager npm (installs dependencies you need to get your front end and back end running)
- APIs (software intermediaries that allow two applications to communicate)
- Callbacks (functions that wait for requests made to endpoints before executing)

- Endpoints (URLs that give APIs access to a requested resource)

The article was written by a content engineer who works for a company called Auth0.

This is ultimately what led me to use Vue.js to develop my front end. The article was super helpful and I used their proprietary login authentication to help secure my application's login process. Auth0 provides the capability of logging in with socials, so rightfully I chose Spotify, which is where the prompt to continue with Spotify comes from. Unfortunately, I still had no idea how to use Spotify's API to make requests, but after some googling, I stumbled upon a Node.js wrapper, which nicely laid out functions that I could use to access different endpoints.

After following the article and slowly piecing things together, I ended up with my front end and functions from the Node.js wrapper that I could integrate into the back end (my API) and use to make calls to Spotify's API. From there I had to ensure that my front end and my back end could communicate effectively.

My front end and back end were able to communicate because of the server.js file. For example, when the top tracks button on the web page is clicked, the mounted function inside of Tracks.vue executes, which calls the getTopTracksFromService function. From there, getTopTracksFromService calls the GetTopTracks function inside of service.js, which waits for a response from my API, api.js. Then, the getMyTopTracks function executes inside of api.js, which hits Spotify's tracks endpoint and passes the response to the getTopTracks function inside of service.js, where it's returned and stored locally inside of Tracks.vue.

Diving into a new framework and using it to try and complete a project focused on something that you have zero experience with over a shortened semester might not have been the best idea. It slowed me down, but I still finished the majority of what I wanted to get done and learned a lot in the process. So overall it was a positive trade-off.

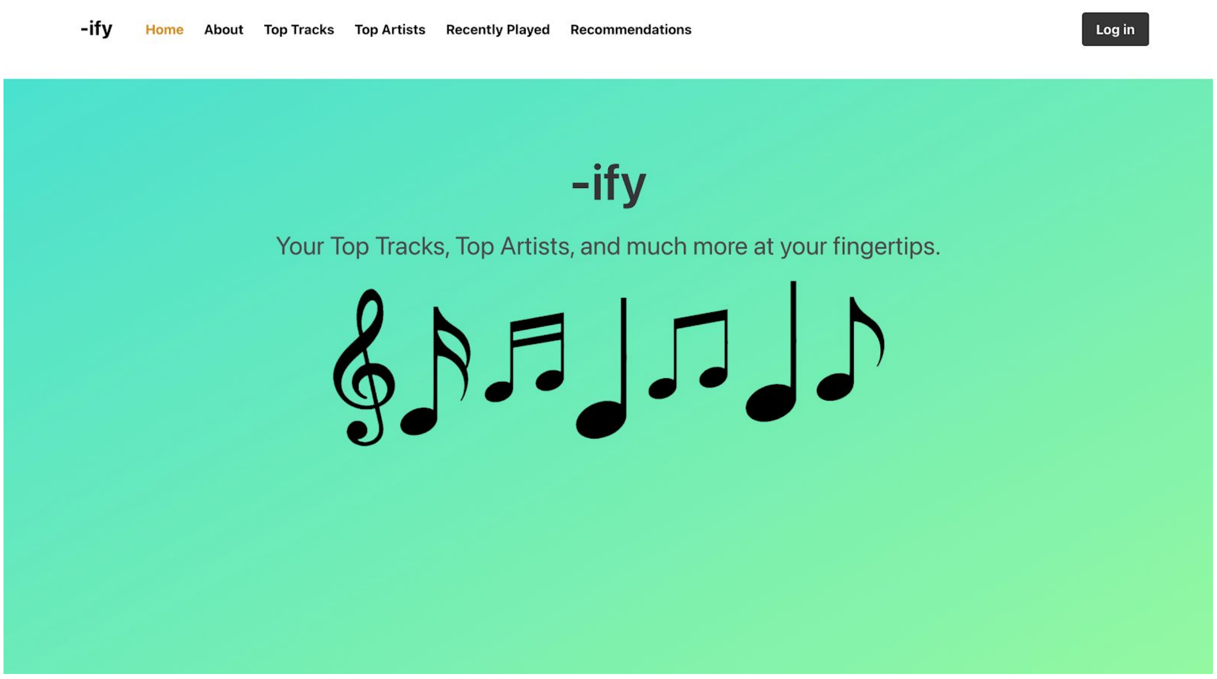
I ran into quite a few smaller issues over the semester and they were mostly things related to my Auth0 account, dependencies, styling, or JavaScript syntax; most of which were a result of my understanding. I also ran into a couple of larger and more time-consuming issues. The first of the two arose when I was trying to generate the authorization through a block of code, and generate it in a way that would not disrupt the flow of the application. This issue became apparent right at the start because I needed a token to test the code myself. I temporarily skipped over it because I could not figure out how to do it. I did not want to waste too much time on it because I could just generate one from Spotify's website. For the first couple of months of development, I would go to the website, generate a token, and copy and paste it into my code and use it until it expired. I would repeat that process whenever I needed a new token. The second larger issue was when I was trying to extract the artists' names, song names, and album art from the JSON response and display it on the front end. It does not sound too hard, but the data that was returned from the endpoints were stored in arrays, and they were all kind of structured differently, meaning I could not just reuse the same block of code four times. I had to loop through different parts of the arrays to extract everything that I needed. it took some time to figure out because, again, I am not very well-versed in Vue.js. I did end up getting everything I wanted, though.

If I were to do this all over again, I would do it in a way that minimizes the use of JavaScript. Although it is such a popular language, I am not too keen on the syntax and it is hard to understand what is going on sometimes. There were a couple of other wrappers that I could have chosen instead, which were written in Python and Java, but I did not know about them when I was starting, nor did I think about any language alternatives because I wanted to experiment with something new. If I had to choose a different language to rewrite the application with, I would choose Python because I have a solid foundation and would not have to spend time learning the very basics. Another reason why I would choose python is because I have already done some development using Java. I prefer python over Java because you can always accomplish whatever you are trying to accomplish with a lot less typing.

I worked hard over the semester and was content with the rate I was progressing. Even though I stayed on track the majority of the time, I would put even more work into the project earlier on while things are not very chaotic. As my other classes started releasing projects near the end of the semester, it became harder to balance my time and I was not able to work on it as much as I would have liked to.

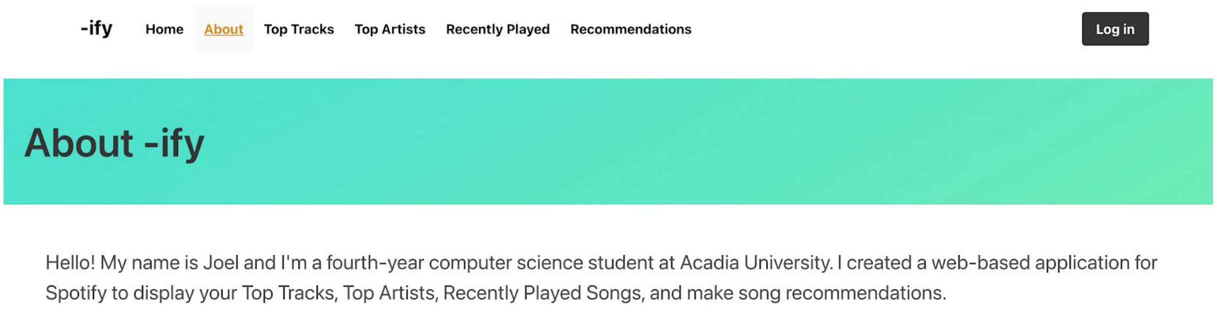
Conclusion

I know that realistically this project could have been completed by someone who knew what they were doing in probably one-third of the time, however, this was my first time working with APIs and doing full-stack development. Overall, I would say that I am mostly satisfied with what I accomplished over the semester. I learned a little bit about what a day in the life of a full-stack developer would look like, a new framework, APIs, and I picked up on countless new programming terms in the process. Although the application is functional, it would have been great to deploy it so that anyone could use it. Here are some final images of the site:



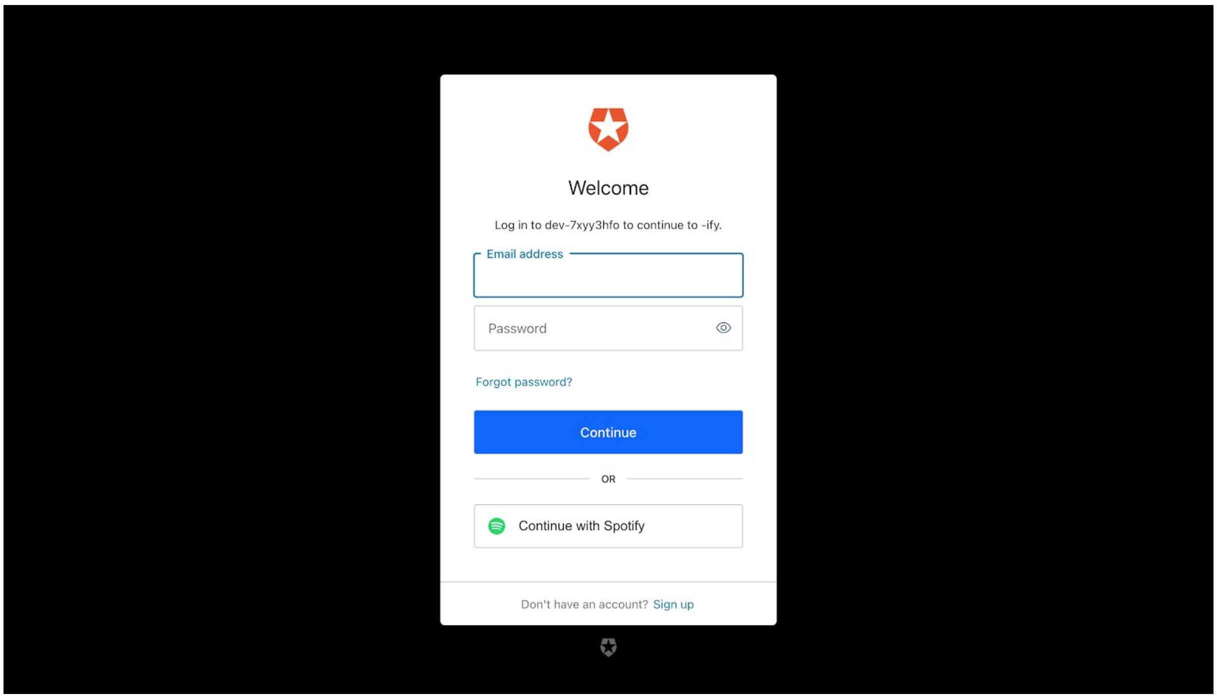
-ify

Your Top Tracks, Top Artists, and much more at your fingertips.

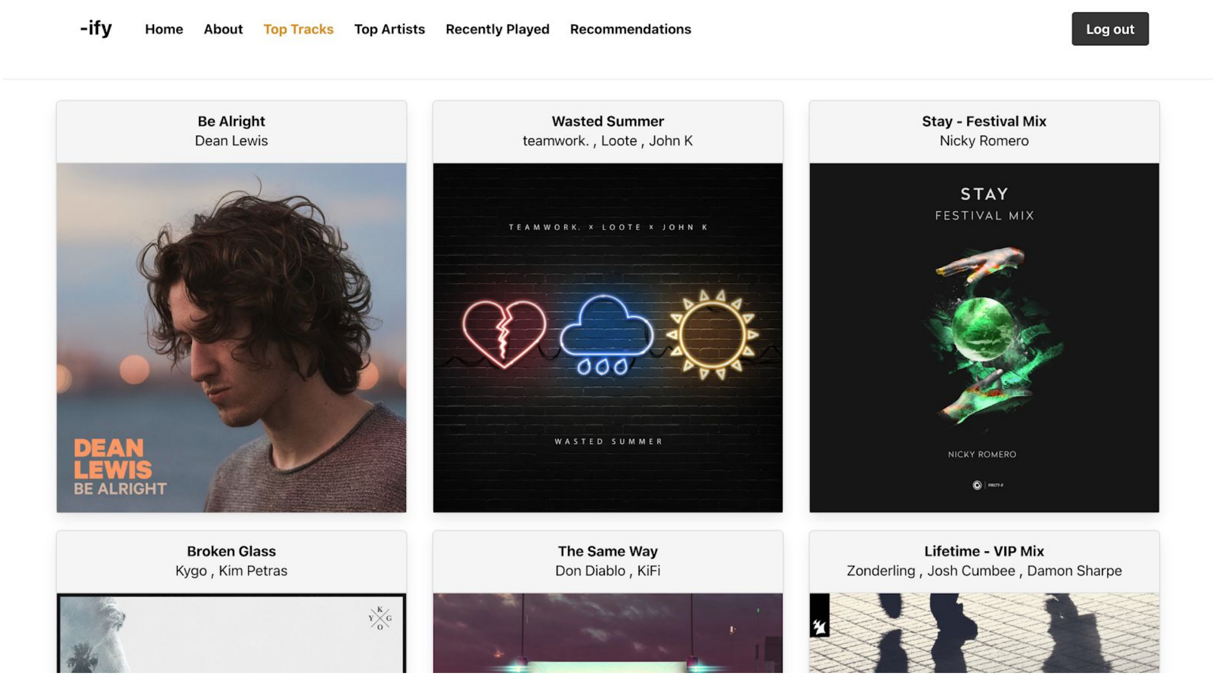


About -ify

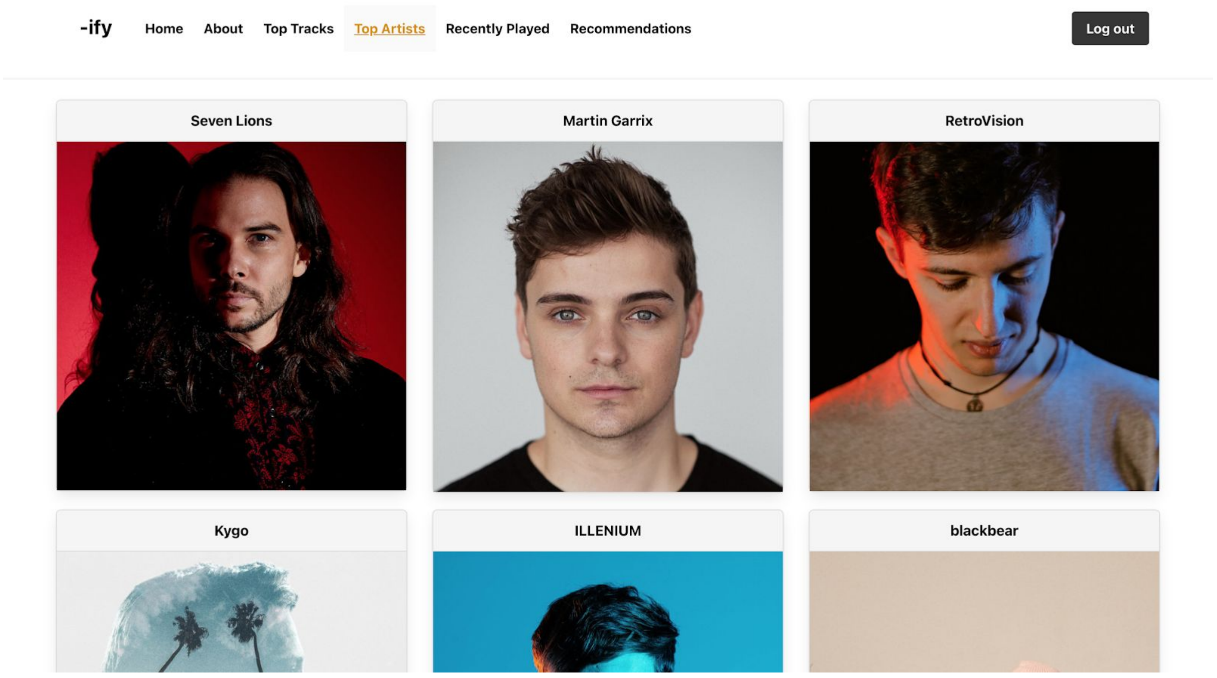
Hello! My name is Joel and I'm a fourth-year computer science student at Acadia University. I created a web-based application for Spotify to display your Top Tracks, Top Artists, Recently Played Songs, and make song recommendations.



Welcome
Log in to dev-7xyy3hfo to continue to -ify.
Email address
Password
Forgot password?
Continue
OR
Continue with Spotify
Don't have an account? Sign up



-ify
Home
About
Top Tracks
Top Artists
Recently Played
Recommendations
Log out
Be Alright
Dean Lewis
Wasted Summer
teamwork. , Loote , John K
Stay - Festival Mix
Nicky Romero
Broken Glass
Kygo , Kim Petras
The Same Way
Don Diablo , KiFi
Lifetime - VIP Mix
Zonderling , Josh Cumbee , Damon Sharpe



-ify Home About Top Tracks Top Artists Recently Played Recommendations Log out

Seven Lions

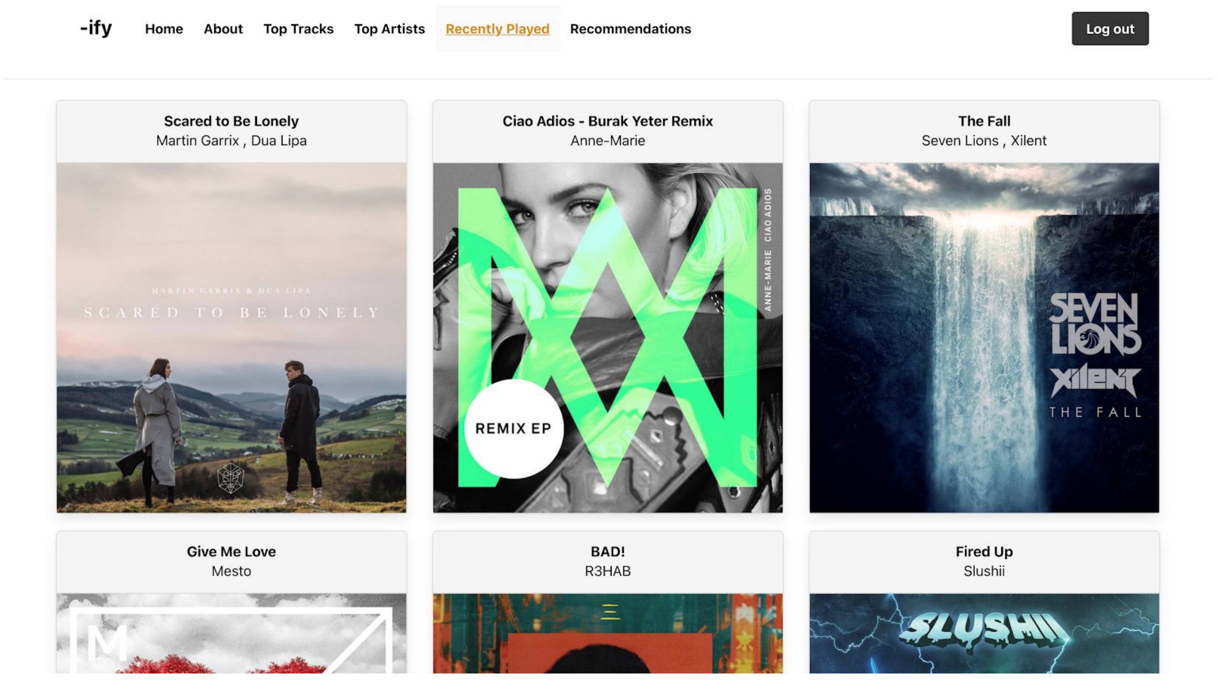
Martin Garrix

RetroVision

Kygo

ILLENIUM

blackbear



-ify Home About Top Tracks Top Artists Recently Played Recommendations Log out

Scared to Be Lonely
Martin Garrix , Dua Lipa

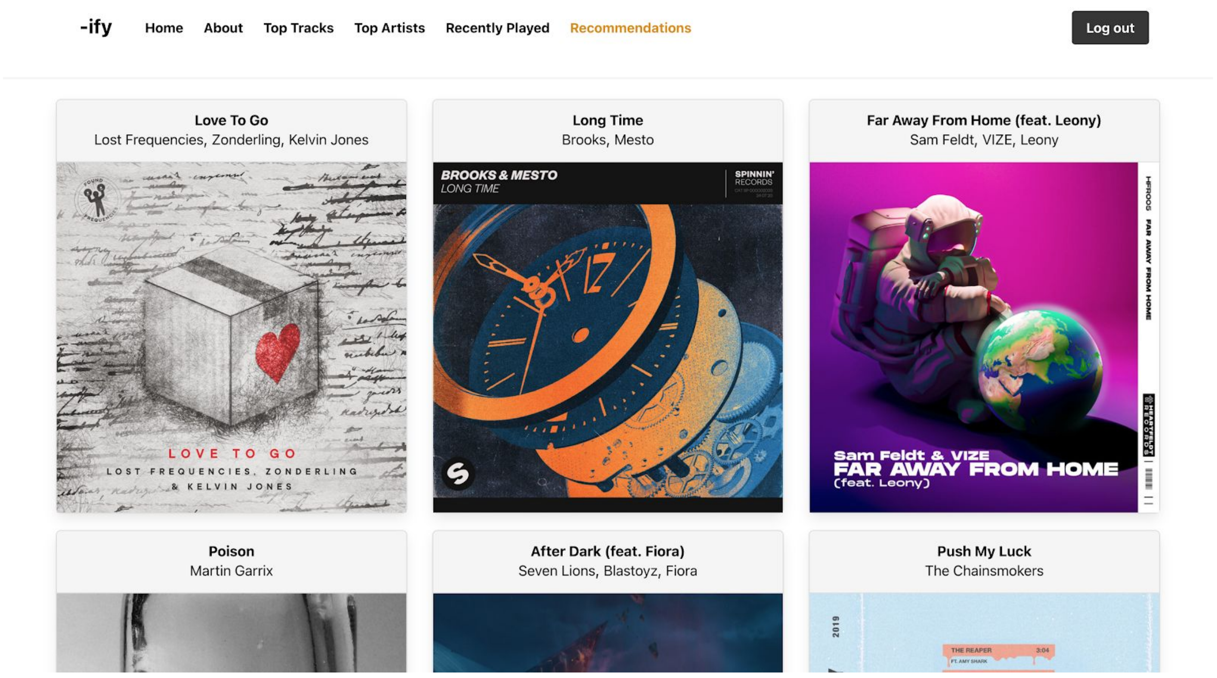
Ciao Adios - Burak Yeter Remix
Anne-Marie

The Fall
Seven Lions , Xilent

Give Me Love
Mesto

BAD!
R3HAB

Fired Up
Slushii



Love To Go
Lost Frequencies, Zonderling, Kelvin Jones

Long Time
Brooks, Mesto

Far Away From Home (feat. Leony)
Sam Feldt, VIZE, Leony

Poison
Martin Garrix

After Dark (feat. Fiora)
Seven Lions, Blastoyz, Fiora

Push My Luck
The Chainsmokers

References & Repo

https://auth0.com/blog/beginner-vuejs-tutorial-with-user-login/

https://auth0.com/blog/how-to-make-secure-http-requests-with-vue-and-express

https://github.com/thelinmichael/spotify-web-api-node

https://developer.spotify.com/documentation/web-api

https://github.com/joeldeighton/spotify-web-app