**Project 3 Team Retrospective**

**Link to website:** [www.museek.us](http://www.museek.us)  
  
**Contact:**   
Alex Pham - [alex.d.n.pham@tamu.edu](mailto:alex.d.n.pham@tamu.edu)

Donald van Steenwyk - dvansteen@tamu.edu

**How to use:** Type in the artist you want to look for in the search bar. The columns on the web page should update with the desired information. Error messages will output as alerts or be displayed inside of the columns themselves. The News API might run out of free requests, so please contact us in that scenario.

**Structure:** We accomplished the minimum we set out to do with our project, which was to integrate three APIs and display their information on our website. Due to a team member leaving unexpectedly, the remaining team members were forced to migrate our entire website onto React, leaving it as a self-contained React application. This created huge problems with integrating our APIs, as the team had to learn React and Javascript in a very short amount of time. In addition, we were forced to change the APIs we were using at the last second, as we found out certain things about React that made it unsuitable for the APIs we were originally using. The upcoming sections will go over each API in individual detail.  
  
**API -   
Songkick:** This API provides information about upcoming concerts that an artist has. This was also one of the first big roadblocks to implementing the website entirely in React, as the form that serves as the search for the entire website had to be moved out of raw HTML and into React, which created problems in transferring data to the API. There were synchronization issues between internal React variables and the information from the API GET request that the team struggled with, but our final product was a formatted list of concerts from the JSON object returned from the API, listed under the “Artist Concerts” portion of the website.  
  
**News API (originally Twitter API):** The Twitter API was our second big roadblock in creating and migrating the API functionality entirely into React. Originally, the team was under the impression that the Twitter API (and Spotify API) required no authentication in order to send GET requests. However, as the team started to implement the APIs in React, we realized that around 2017, Twitter and Spotify started mandating API authentication with OAuth 2.0.

This led to an attempt to implement OAuth 2.0 in our API calls, and then realizing that OAuth 2.0 was restricted to purely backend Javascript, or server side calls. React exists as a purely frontend Javascript framework, which essentially meant that it was impossible to implement Twitter/Spotify API calls in our React application as is, unless we created an entire backend for our React app as well. This created a new problem, as we were hosting the website on AWS, originally using EC2, but later migrating to S3, which was using static website hosting on a serverless model. Even if we were to spend the time learning how to make client and server communicate with each other, we would still need to figure out how to host both on AWS and have them communicate on there.

Ultimately, the team decided to scrap any API with authentication, and to replace the Twitter API, we went with News API, an open source news API that returned a JSON object that we formatted into a list of relevant news articles about the searched artist under the “Artist News” column of the website.

**Last.fm (originally Spotify API):**

Because Spotify had similar issues to Twitter, we switched to Last.fm to provide the website with music. The Last.fm API provides information about artists their albums, songs, and popularity. It also provides links to events and concerts, but Songkick already provided those to use. Using links given by Last.fm API calls, users can listen to albums and songs linked through the website. We used fetch calls to retrieve data inside of a React class and used the render function to return HTML to the app. It was not as convenient as using something like a pre-built wrapper like Spotify had, so it took extra work to get it to work properly.  
  
**User Evaluation:** Our user evaluation was only useful for design of the website. With the team having to implement the APIs at the last second, we were forced to redesign the entire website/application from the ground up anyway, making any information outside of design useless.

**Contribution:**

Alex Pham - Implemented the News API and Songkick API, migrated HTML/CSS into React, wrote report (except for Last.fm section)

Donald van Steenwyk - Implemented the React app backbone, implemented the Last.fm API connection, wrote Last.fm section of report  
Joseph Orosco - Video

**Grade Multipliers:**Alex Pham - 1.3  
Donald Van Steenwyk - 1.1  
Joesph Orosco - 0  
  
Donald and I have agreed on these grade multipliers. Joseph did the video, but we have been unable to contact him, and his portion of the project (the APIs) never materialized, so Donald and I did his portion as well. While he did do the video, the multipliers were assigned primarily on a code basis, and Donald and I have actually yet to see him in person since Project 3 started.