**CSEE 5590 - 003 Project Report**

****

**Project Title:** Tune Link

Group 3 Team Members

Joe Moon; jmn5y@umsystem.edu

Nathan Cheney; ncxn8@umsystem.edu

Keenan Flynn; kpfxn8@umsystem.edu

Jasmine Naraine; jnytc@umsystem.edu

[Video Demonstration](https://youtu.be/0Axdd8SoQ3s)

Introduction & User story

TuneLink is an online, interactive concert search engine that uses a simple API to find concerts that are relevant to a user. As live music enthusiasts, we want a place to view concerts that might be interesting to us. Our search algorithm factors in personal music preference to find vibrant local events. Several online services have concert search engines, but each comes with advantages and drawbacks. TuneLink seeks to solve some of the problems that we’ve identified with these competitors.

We’ve identified in-person concerts as a resurging market. Covid-19 originally eliminated the possibility for these events, but with lockdowns ending TuneLink is primed to fulfill user needs. Users are able to search by 3 primary categories: Spotify ID, Location, and Artist.

With TuneLink, the user is able to enter a small amount of information about themselves (ie Spotify userID) and receive a list of events that are relevant to our music preference. When a user selects one of these events, we would like to display information about that event such as the location and time. When a user clicks on the location, we would like the user to be directed to a site which contains more information about that location.

Background/Related work

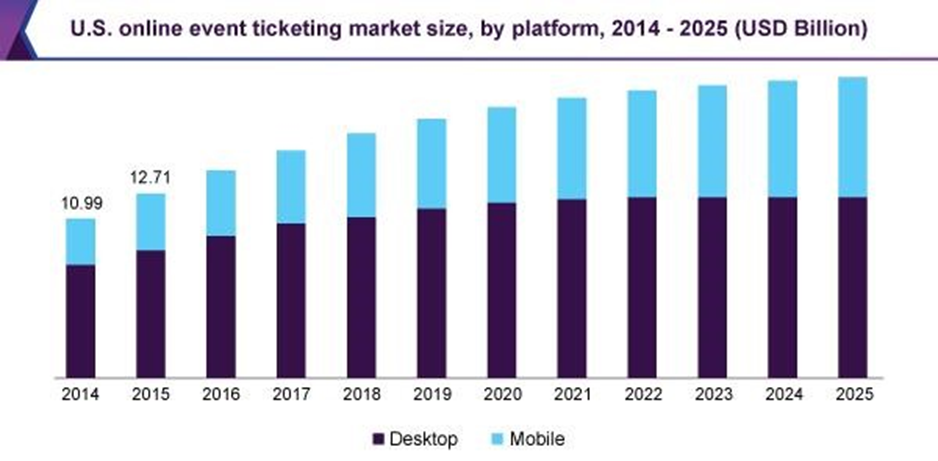
Our project has 2 main competitors, [Spotify](https://www.spotify.com/) and [SongKick](https://www.songkick.com/).

Spotify is a premier music streaming service, Spotify allows users to create playlists and follow artists. The Spotify algorithm creates tailored suggestions to the user which allows them to discover new music and keeps them coming back for more. Spotify does have a concert interface where users can find interesting concerts but this functionality is hidden behind several layers and actions and the interactivity is limited (no artist search).

SongKick is a concert based online service. Its flagship service is a concert finding search engine. This engine is robust and can return artists, venues, locations, and more. SongKick needs the user to input specific search criteria/artists and cannot ingest lists into the search. SongKick has some capability to search by Spotify ID, but this interface requires the user to log into their Spotify Account through SongKick.

Other apps, such as [BandsInTown](https://www.bandsintown.com/) have concert finding algorithms that incorporate a Spotify search. To use this capability for BandsInTown, the user has to create an account and do a 3rd party login to Spotify. This poses security risks for the user. The more often a user enters their password through different vendors, the weaker that password becomes. TuneLink only uses parts of the user's Spotify profile that are publicly available so that the user does not have to jump through these extra hoops. More about this is explained in the Methodology section.

We have also identified a business need for TuneLink. Online ticketing sales has been an increasing market over the last 20 years. (46.59b market in 2017). Physical music media (CDs/Vinyl) markets have dropped from $13.2 billion in 2000 to $1.65b in 2021 with streaming music comprising 83% of the market. Large players currently exist in both the online ticketing and streaming markets, but none merge data to make smart recommendations for concerts.



Proposed idea

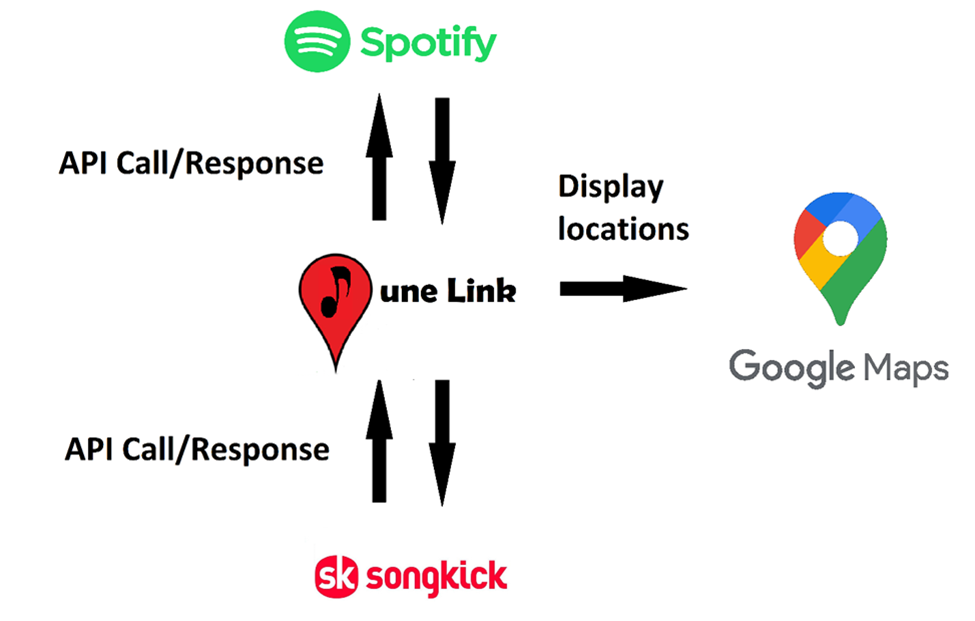
With TuneLink we want to bridge the gap between the Spotify and SongKick APIs. We’d like a user to be able to enter their Spotify ID, and have the robust SongKick API return a list of relevant concerts in the area. We also want the user to be able to search by location and by artist. We want to package these searches into a clean and simple UI so that the user does not need to dig through several screens to get their desired output. Finally we’d like to be able to connect to the Google Maps API so that a user can be given directions to their event although this is not yet implemented as on 4/11/2021.

Methodology/Approach

We have implemented TuneLink using Angular and RESTful APIs. TuneLink communicates with both the Spotify and SongKick APIs. When a user wants to search for concerts, they are able to enter their Spotify User ID. This ID is then used to query the Spotify database and GETs a list of artists from that users created playlists. User created playlists are public by default on Spotify. Because this information is public, we do not require the user to log into their account. This is where TuneLink differs from other concert search algorithms which do require a 3rd party login.

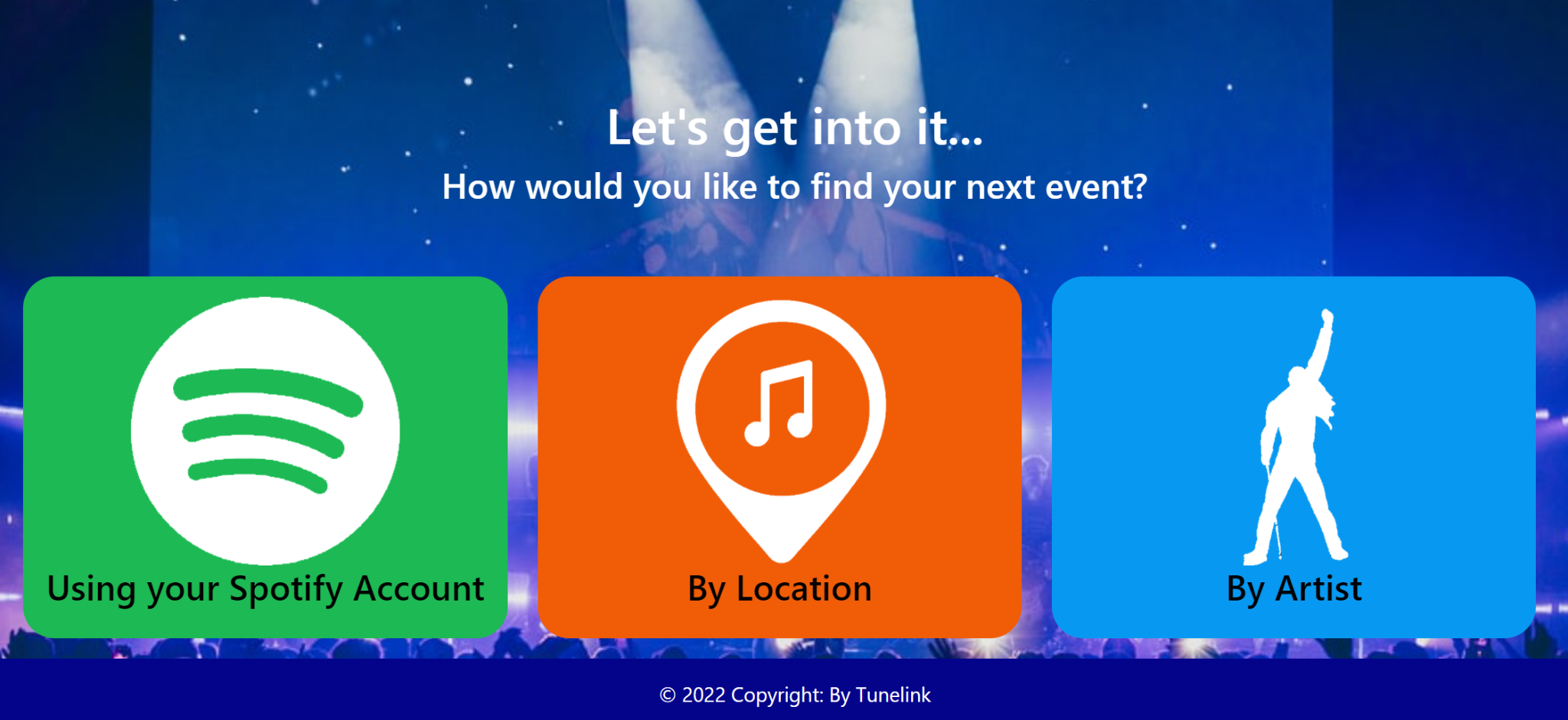
With the list of artists we then query the SongKick API to get any available concert information. In this way we are tailoring the results to fit the users tastes. The results have links to the venue and event that will direct the user to utility information about the venue such as directions and opening/closing times.

The project is broken up into over a dozen different Angular modules. There are modules for the navbar, artist search, location search, spotify search, results and more. These modules communicate with each other to get relevant results. We have also included error pages, a nav bar, footer, and search pages. The user interface is implemented with HTML and CSS.

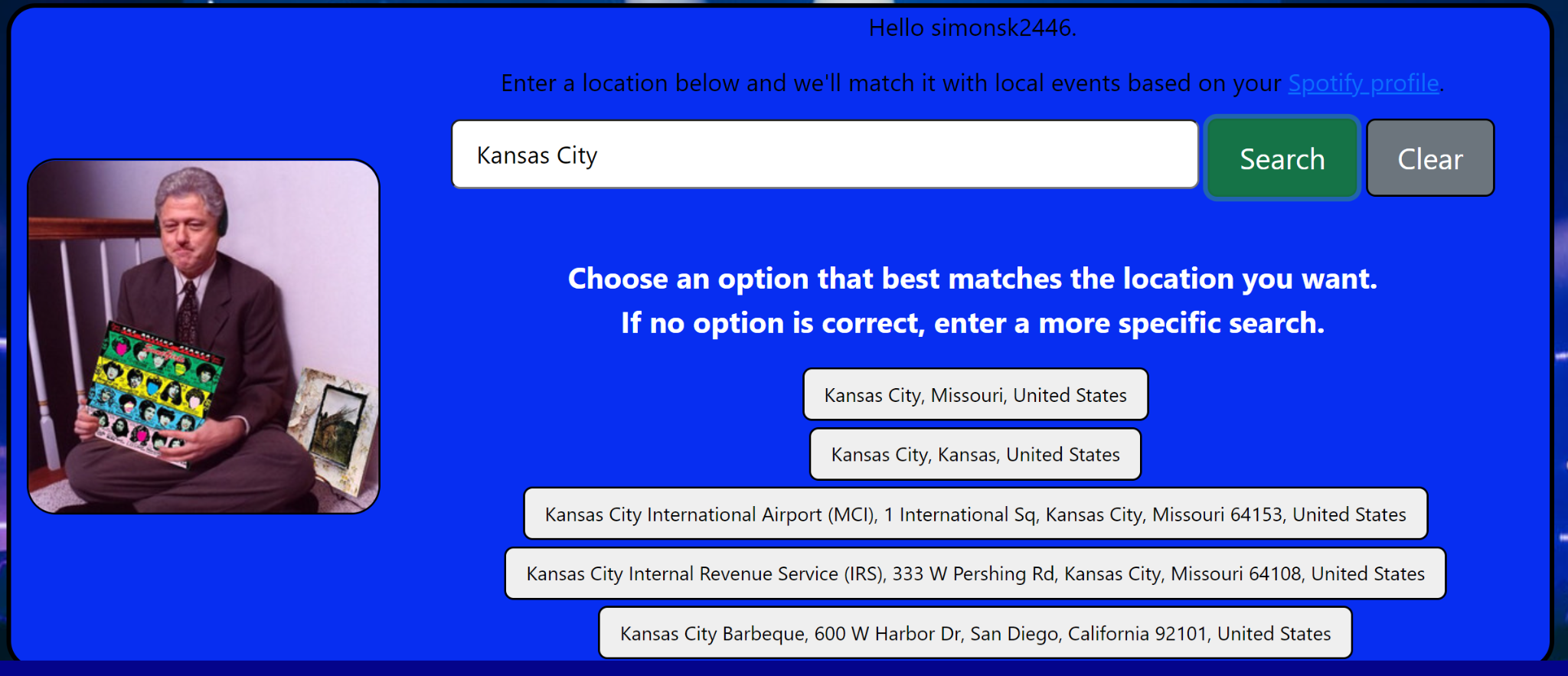


Features

The primary feature of our app is a concert search engine. This search engine is delivered through 3 distinct searches. The user can search by Spotify User ID, location, and artist. There is a simple UI design that directs the user to the relevant page for each search option.

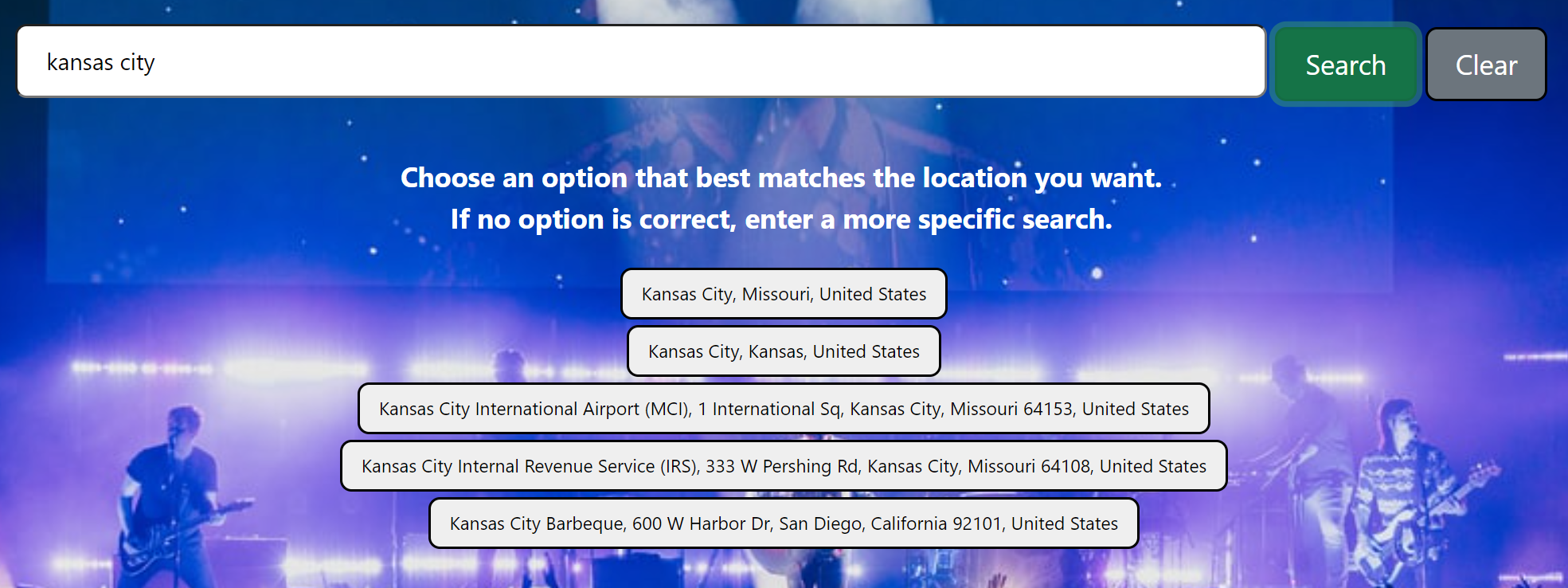


The Spotify Account search asks for the user's Spotify ID. It directs the user to a new screen where it shows the profile picture for that Spotify account and asks the user to enter a location. The algorithm will then return concerts in the area that are relevant to that Spotify account.The user can click on the Event link and be directed to the SongKick page specifically for that event. They can also click on the Venue link and be directed to the SongKick page specifically for that venue.





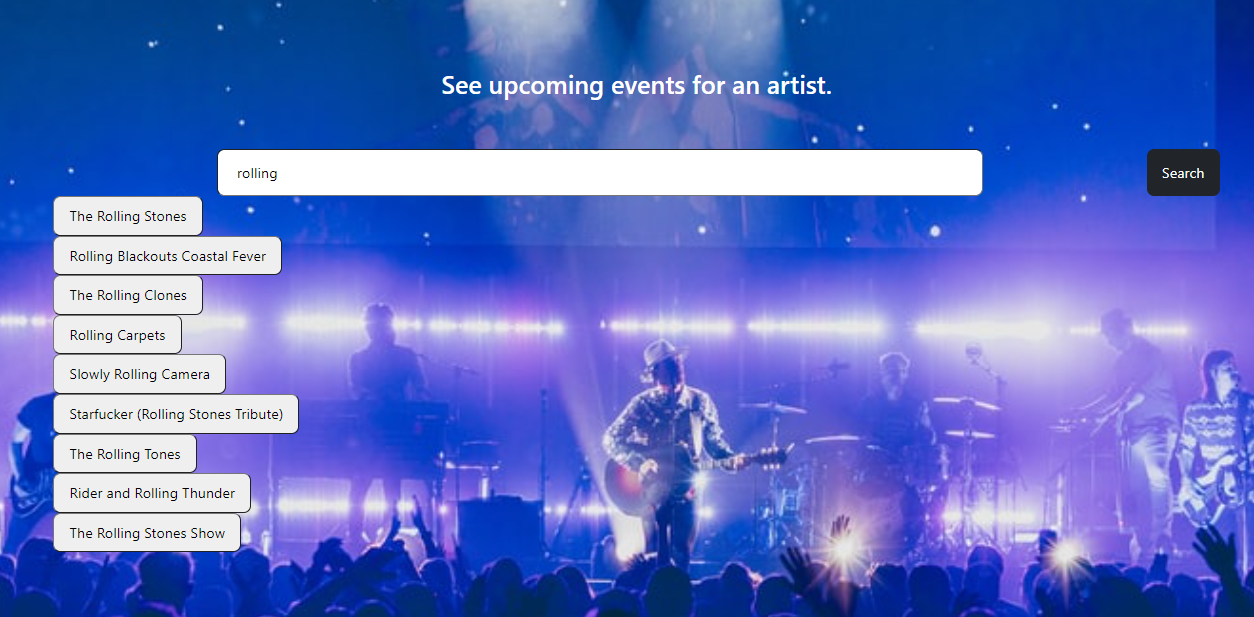
The location search asks the user for a location such as a City, State, Country or Venue. If you enter a broad query, the engine asks you to further specify your search by showing a list of relevant locations. It then connects to the SongKick API and returns concerts that are listed to happen in that specific location.



Again the user can click on the Event or Venue link and be directed to the corresponding SongKick page.



The artist search asks the user to input a specific artist. Once an artist is entered it will bring up a list of artists that match that user's input. For example, if a user input the word ‘rolling’, the search would return a screen that looks like this.



The user would then be able to select the artist ‘The Rolling Stones’ and be shown a list of upcoming concerts that look like the following.



Results. Evaluation & Conclusion

TuneLink fills a gap in the concert market. Concert goers are not always tech savvy, and TuneLink offers a streamlined way to browse for concerts.

TuneLink shines in that it requires no 3rd party logins into Spotify. This generates no security risk for the user and allows them to enter the absolute minimum amount of information about themselves. Users can often be lazy, and when they are presented with applications that take work on their part (such as importing their music library), they will most likely find alternative solutions.

TuneLink is one of these alternative solutions. Users only need to enter a minimal amount of data, such as a Spotify ID or a location and be connected with meaningful concerts. Though we still have work to do, such as implementing Google Maps, we are proud to present TuneLink.

Improvement from Increment 2

Since Increment 2, the following changes have been made along with other enhancements.

* Integration of SongKick API
* Expanded search into 3 categories: Spotify ID, Location, and Artist
* UI overhaul and enhancements
* Presentation creation

Contributions

* Joe Moon: Implementation of APIs, creation of Angular app
* Jasmine Naraine: UI/UX, design of HTML and CSS
* Nathan Cheney: Documentation and information gathering for report
* Keenan Flynn: Presentation and Report contributions, Research on Songkick API

Project GitHub Link

* <https://github.com/WebProject-CSEE5590/project/tree/master>

Video Link

* <https://youtu.be/0Axdd8SoQ3s>

References

* API for music information: <https://developer.spotify.com/documentation/web-api/>
* API for concert and festival listings: <https://www.songkick.com/developer>
* Business info about concert sales <https://www.thedenverchannel.com/news/national/cd-sales-see-resurgence-as-overall-physical-music-sales-grew-for-first-time-since-1996>

<https://www.grandviewresearch.com/industry-analysis/online-event-ticketing-market>