Tunes

Team Members: Kevin Alarcon, Munkh-Erdene Baatarsuren, Kyle Godinho, Damain Moquin, Evan Moore, Jack Nguyen

Spring 2018

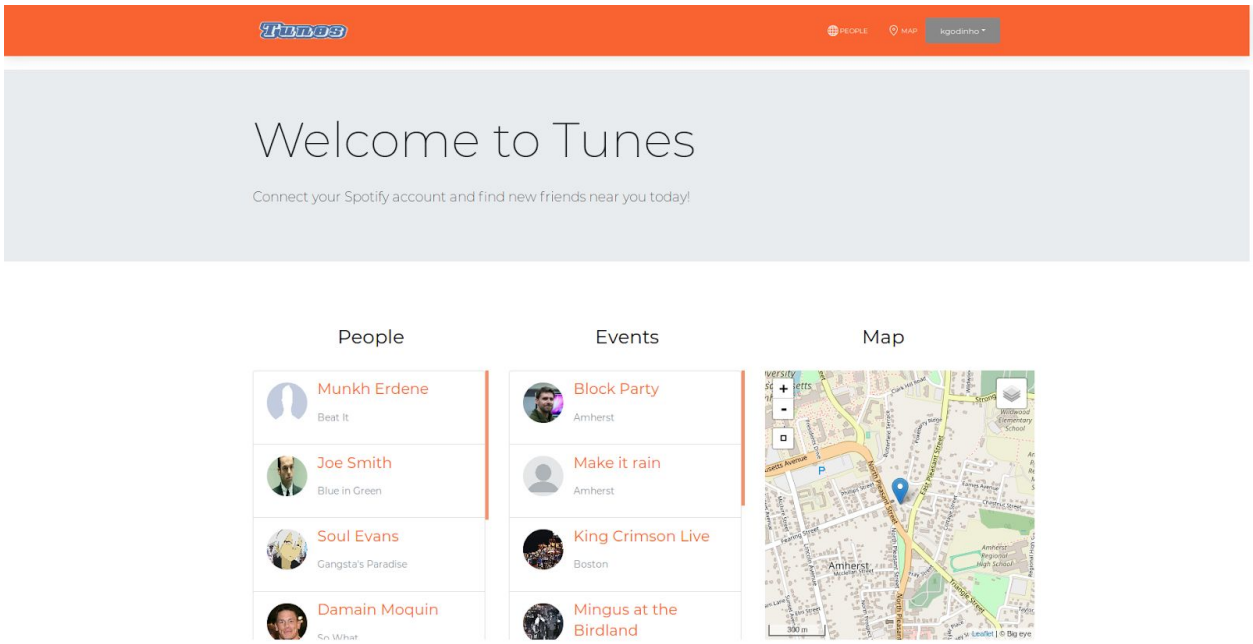
[Github]

Overview

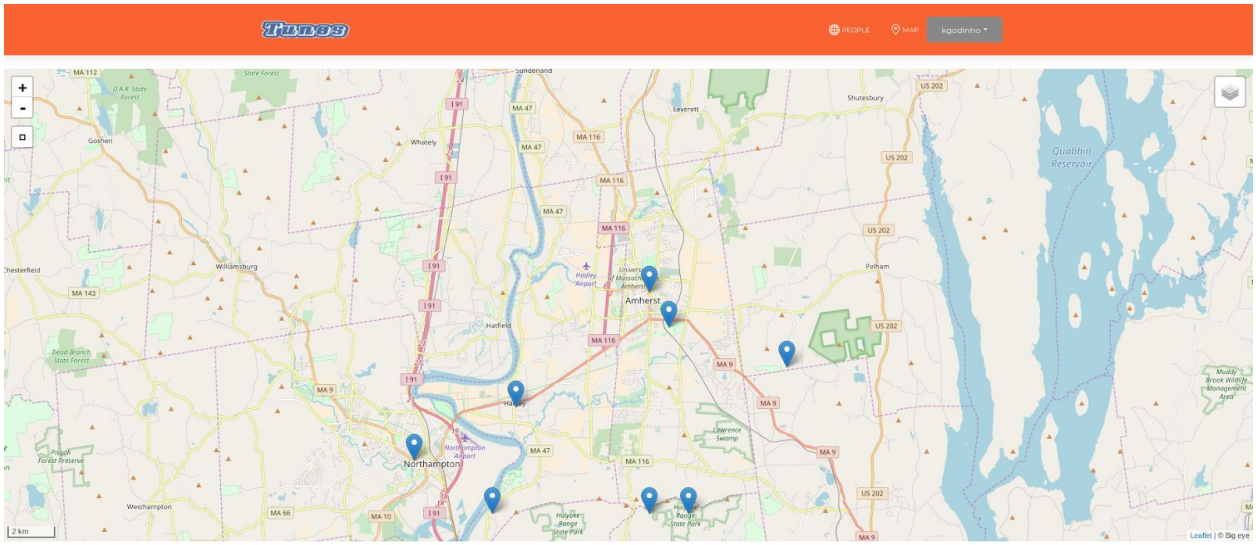
Tunes is a social media platform with a focus on connecting nearby users through a shared interest in music. The application uses your location to show your friends and others what's in your rotation and what events you're going to. Want to go to the Cardi B concert but don't have anyone to go with? Tune in and find someone else who wants to go! Have you ever had a friend who always gets asked "what song is this?" when they have the aux cable? Tune in to their profile and see what they're listening to, their current anthem--which says a lot about them, and what shows they've been to. With the addition of our interactive maps, you'll be able to meet others who share your musical interests. Our platform provides opportunities from finding that last member for your band, to promoting a party you're throwing at your house, to meeting your new best friend who might be the only person who likes Frank Ocean more than you do. Other sites exist for tracking your musical listening history, but none are as integrated with social media as Tunes.

User Interface

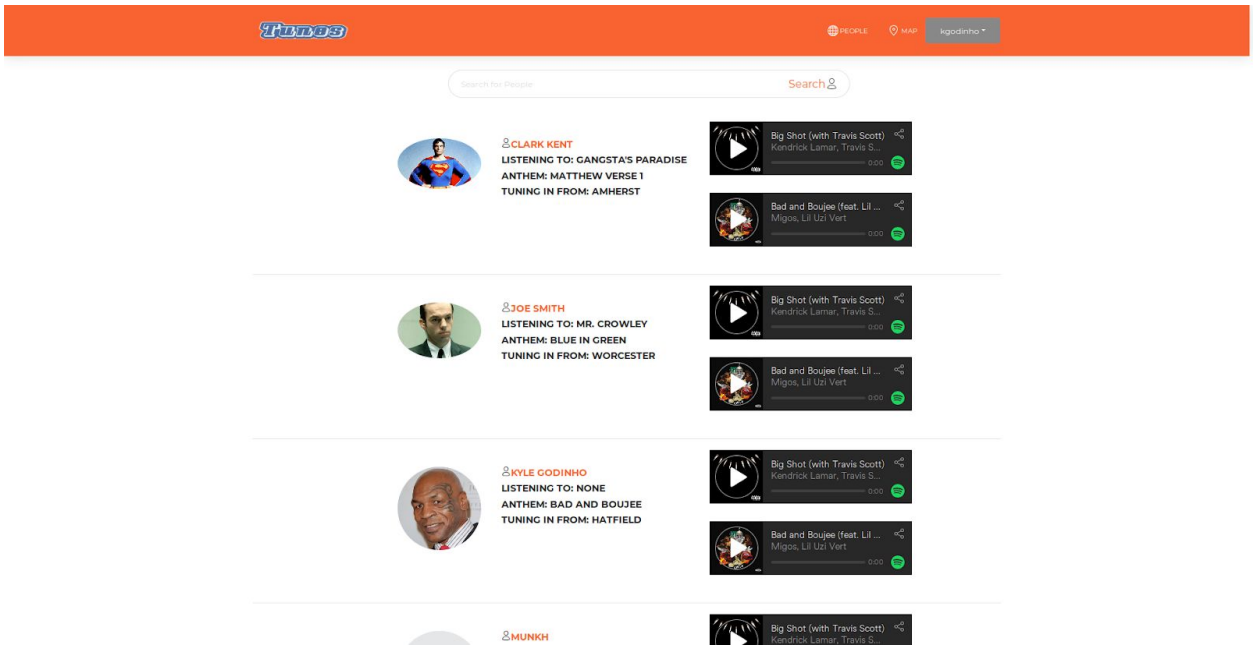
Home Page: Application's landing page. Displays other users, events, and the map to the user on single page in a simple format.



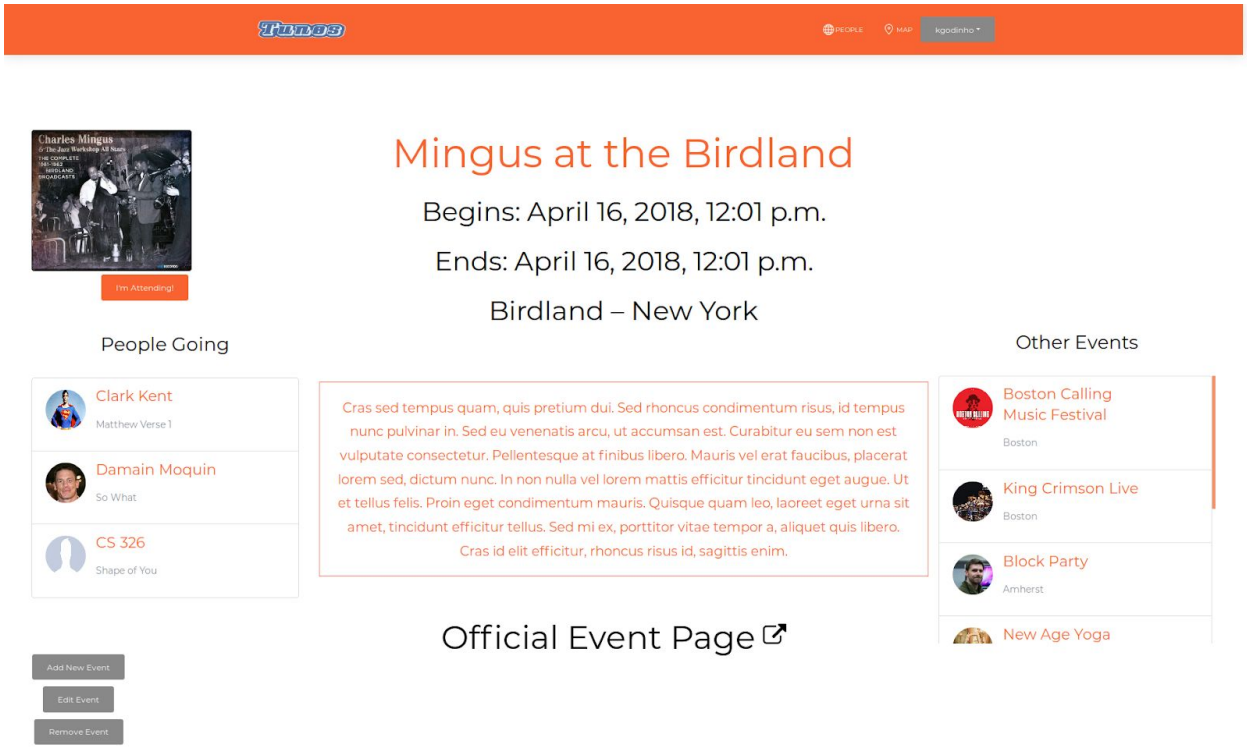
Map Page: A larger and more easily viewable map that the user can interact with.



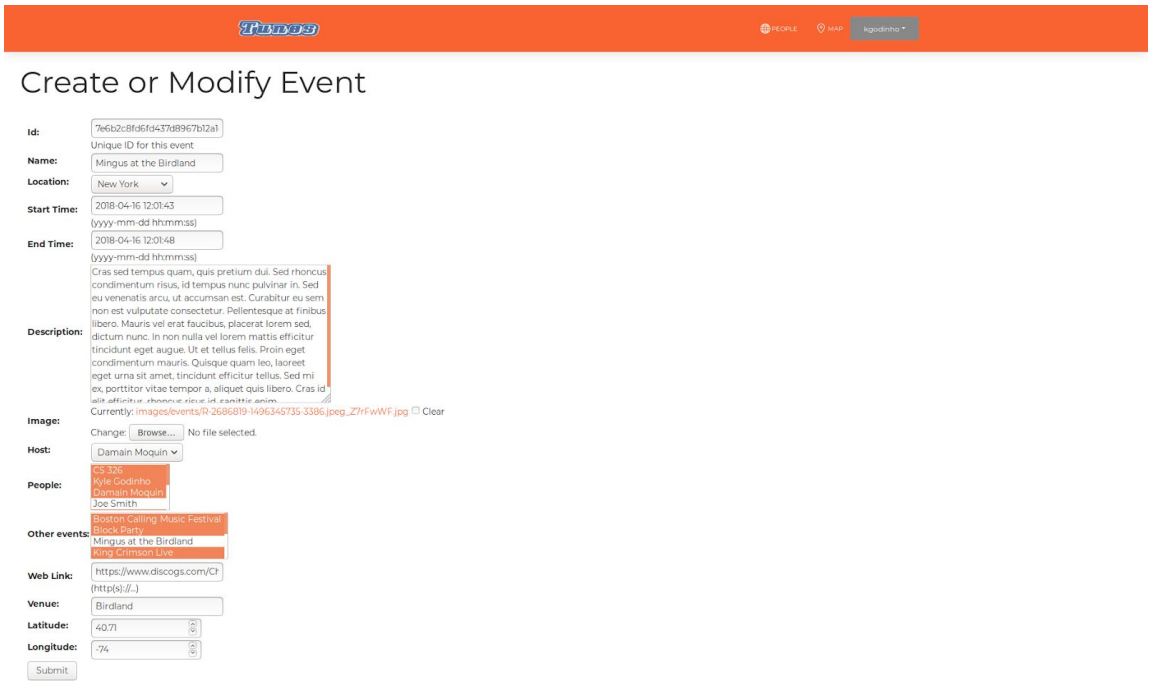
People Page: A list of other users showing more information than the home page and playlists that they like.



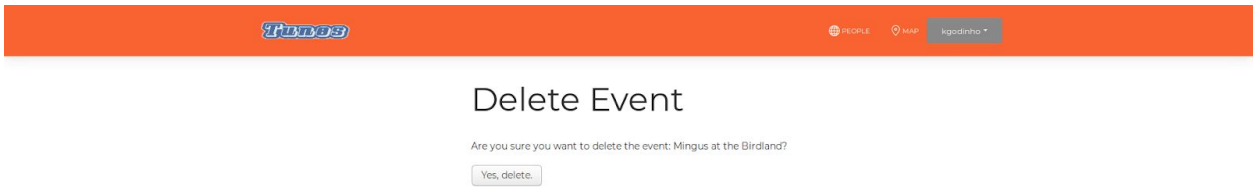
Event Page: A page individualized for each event. Shows the event details and displays who is going along with other events. Gives a user the option to add, edit, or remove the event if they are an event admin.



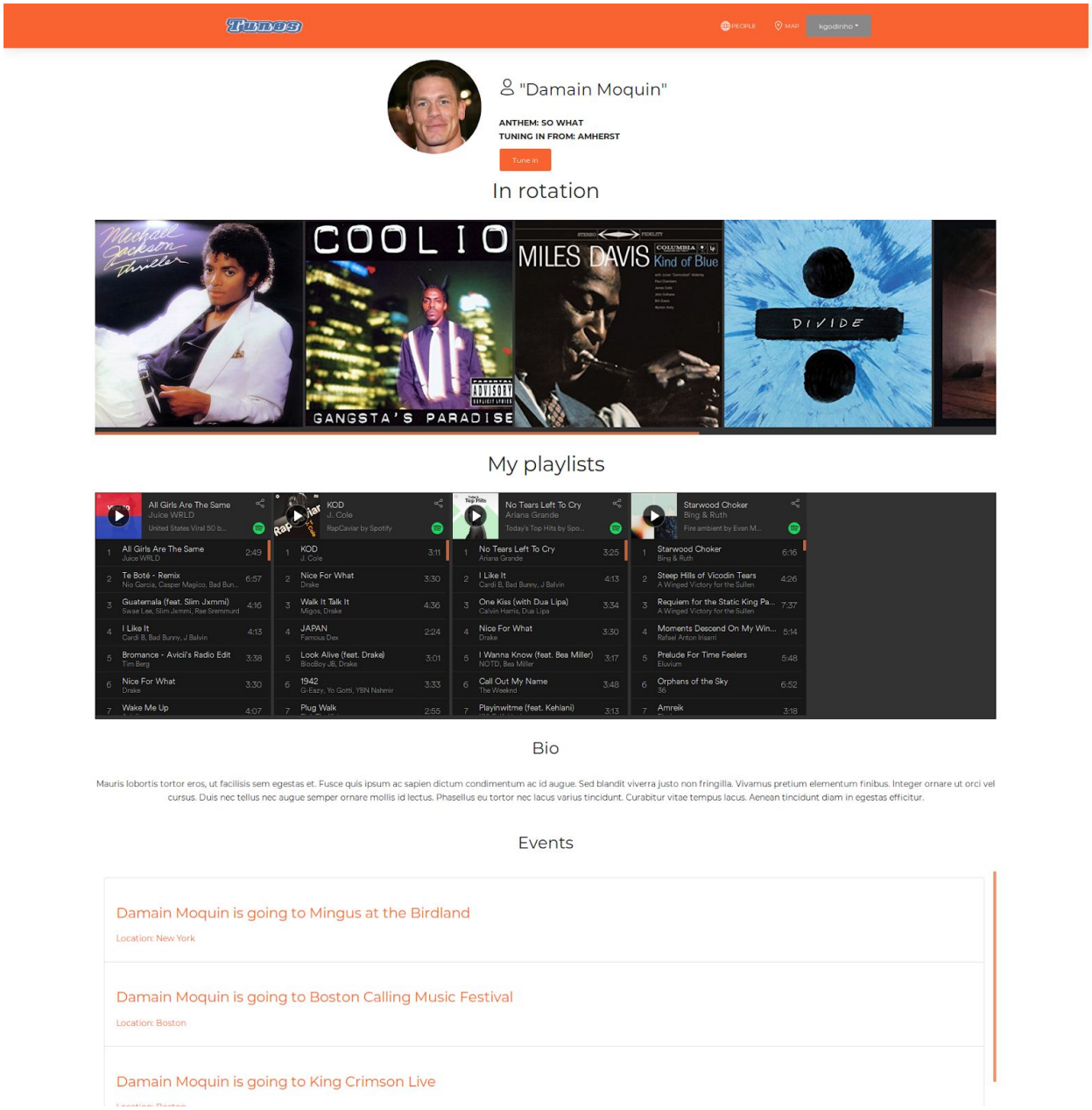
Event modify/create Page: A page used either for creating a new event or modifying an existing one.



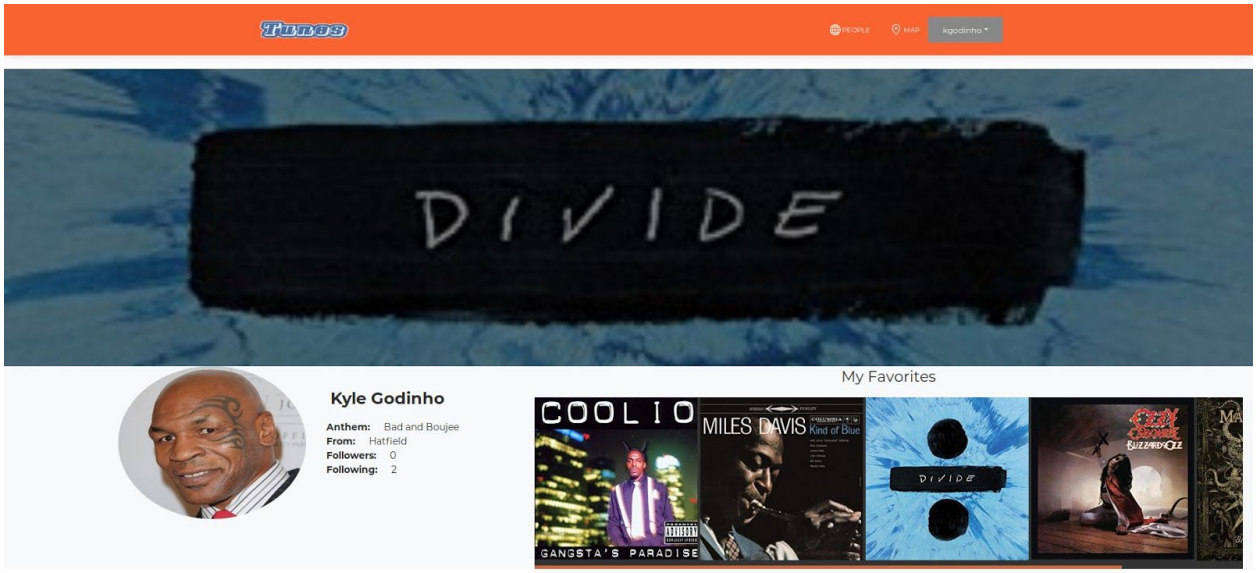
Event remove Page: A confirmation screen after selecting “Remove Event” before the Event is removed.

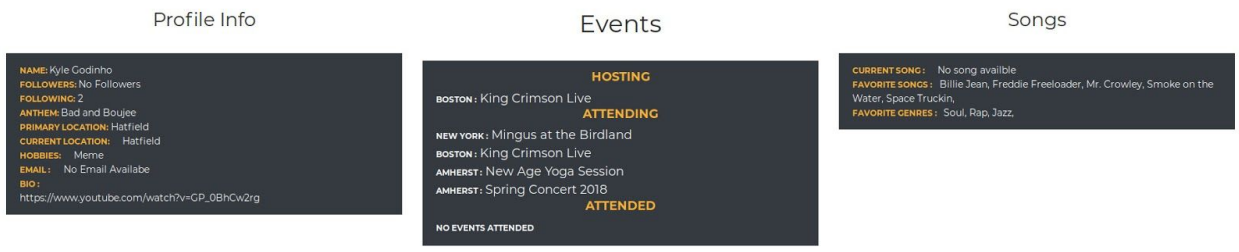


Profile Page: Individualized page for a user’s profile page. Displays their information such as their music rotation, bio, and events attending.



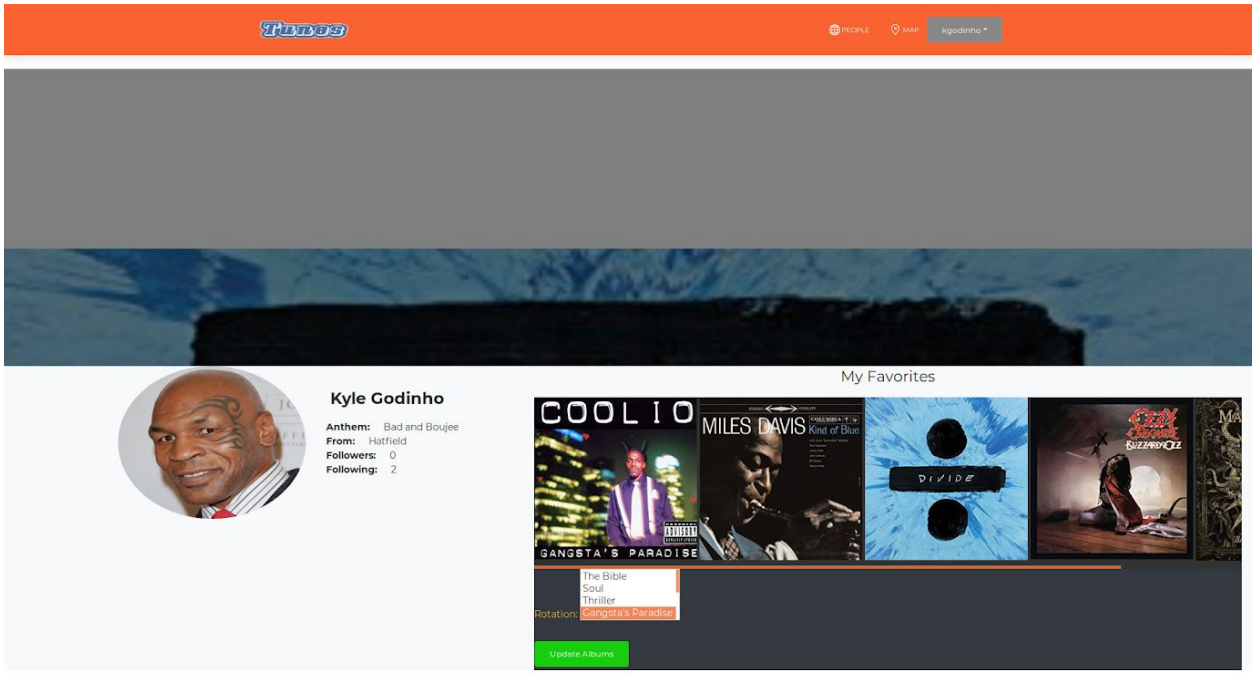
My Profile Page: Displays the user's own profile information in a compact and easy to read format.

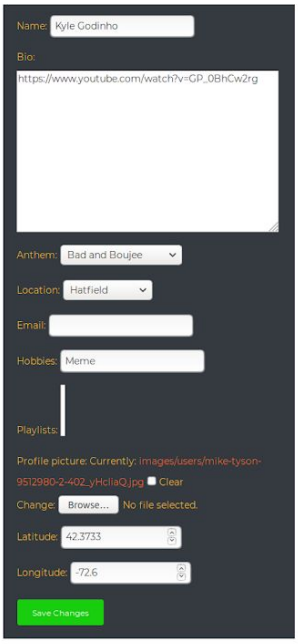


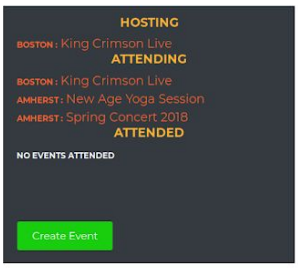


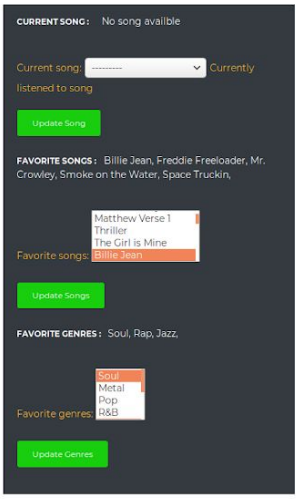
Login Page: The Login page where new users can also select a link to create an account.

Settings Page: Similar to the “My Profile” page but allows the user to modify their information.

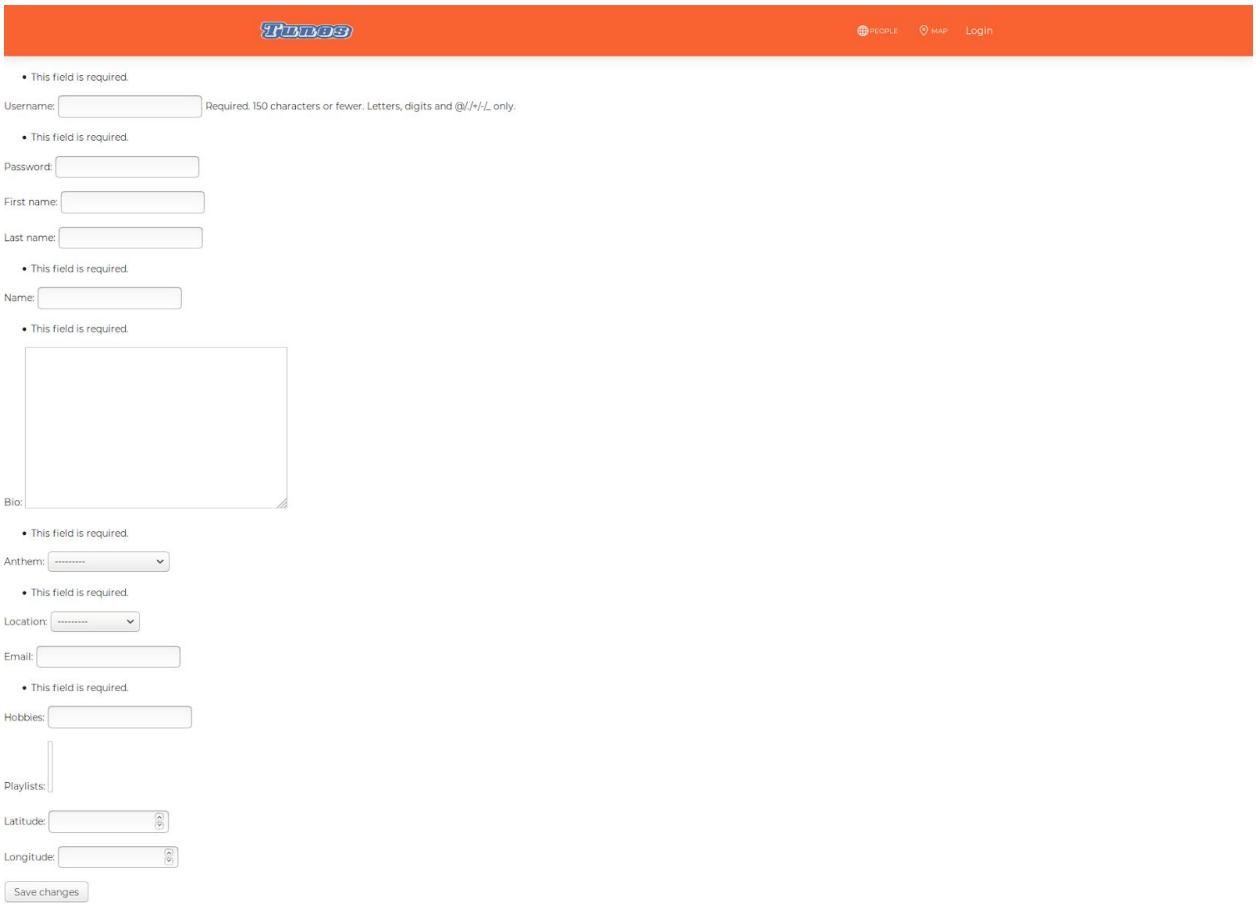




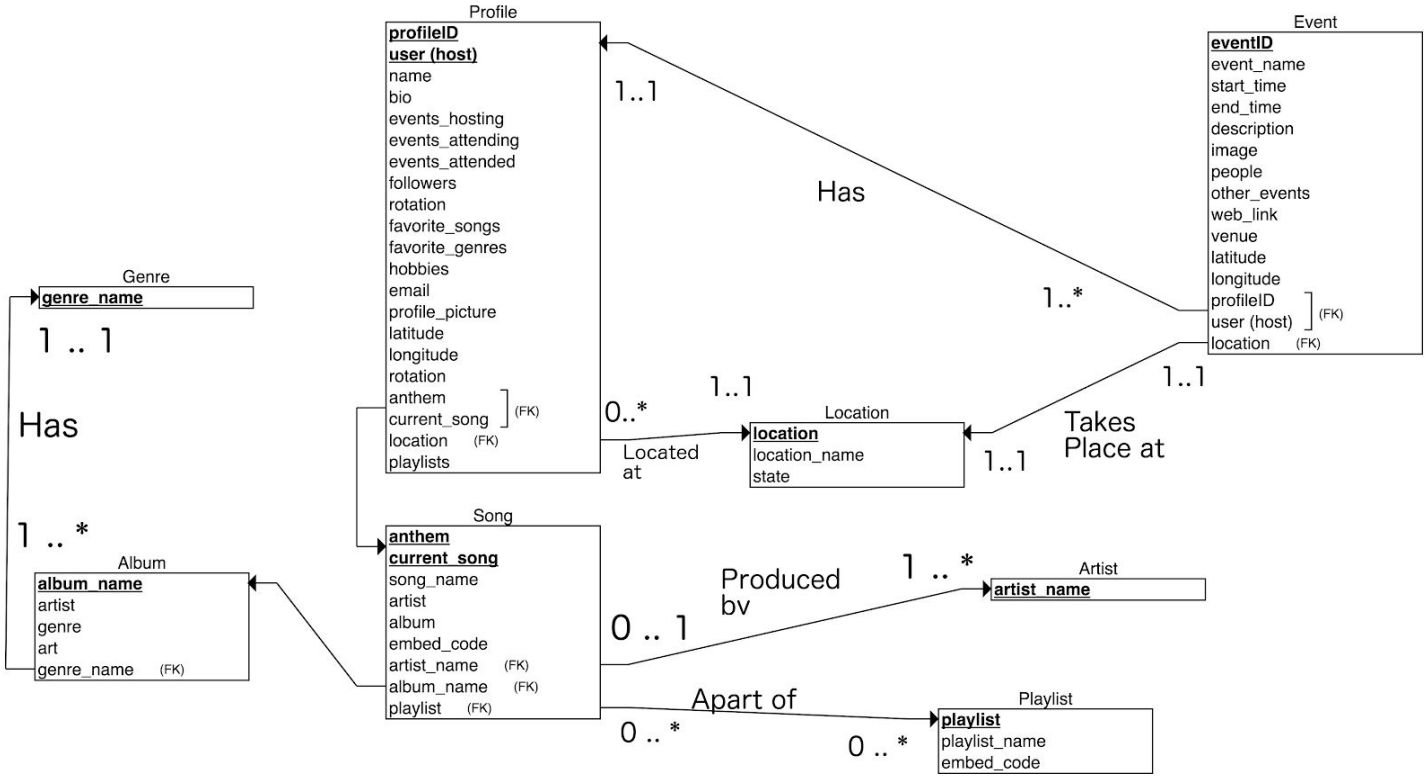




Create User Page: The page where a new user will create their profile and add data to their profile during creation.



Data Model



The profile model is our most important model and includes a distinct profile identifier for each user of the site. Each profile contains attributes such as the user's name, bio, which events they're attending and hosting, their email, location, along with other personal information. It also includes foreign keys from other models like anthem, current song and location. Anthem and current song relates each profile to the song model, telling us what the users' favorite song is and what they are currently listening to. The song model links the album, artist and playlist model, each of which are made for ease of adding to the database. The location key belongs to the location model and tells us where each user is on the map. This field is also contained in the event model and tells us where each event is on the map. Events are linked to the user profiles through a one-to-many correspondence where each event can only have one host, but users can host many events.

URL Routes/Mappings

URL Route/View	Description	Authentication
Tunes/, views.index	Links homepage to default web address of site.	None.

Tunes/user/userID, generic user page view	Regex search to link to each user page from homepage.	None.
Tunes/event/eventID, generic event page view	Regex search to link to each user page from homepage.	None.
Tunes/maps, views.maps	Links to full map view from homepage or nav bar.	None.
Tunes/my_profile, views.my_profile	Links to the currently logged in user's profile page from nav bar.	Only available if user is logged in to an account.
Tunes/settings, views.update_profile	Links to the currently logged in user's settings from nav bar.	Only available if user is logged in to an account.
Tunes/people, views.people	Links to the people view from the homepage or nav bar.	None.
Tunes/profile/create, views.create_profile	Links to the create profile form from the log-in page.	None.
Tunes/event/create, generic event creation view	Links to the create event form from the event or settings page.	Only available if the user has an account within the "Event Admin" group that is logged in.
Tunes/event/eventID/update, generic event update view	Links to the event update form from the page of that event.	Only available if the user has an account within the "Event Admin" group that is logged in.
Tunes/event/eventID/delete, generic event delete view	Links to the event delete form from the page of that event.	Only available if the user has an account within the "Event Admin" group that is logged in.

Authentication/Authorization

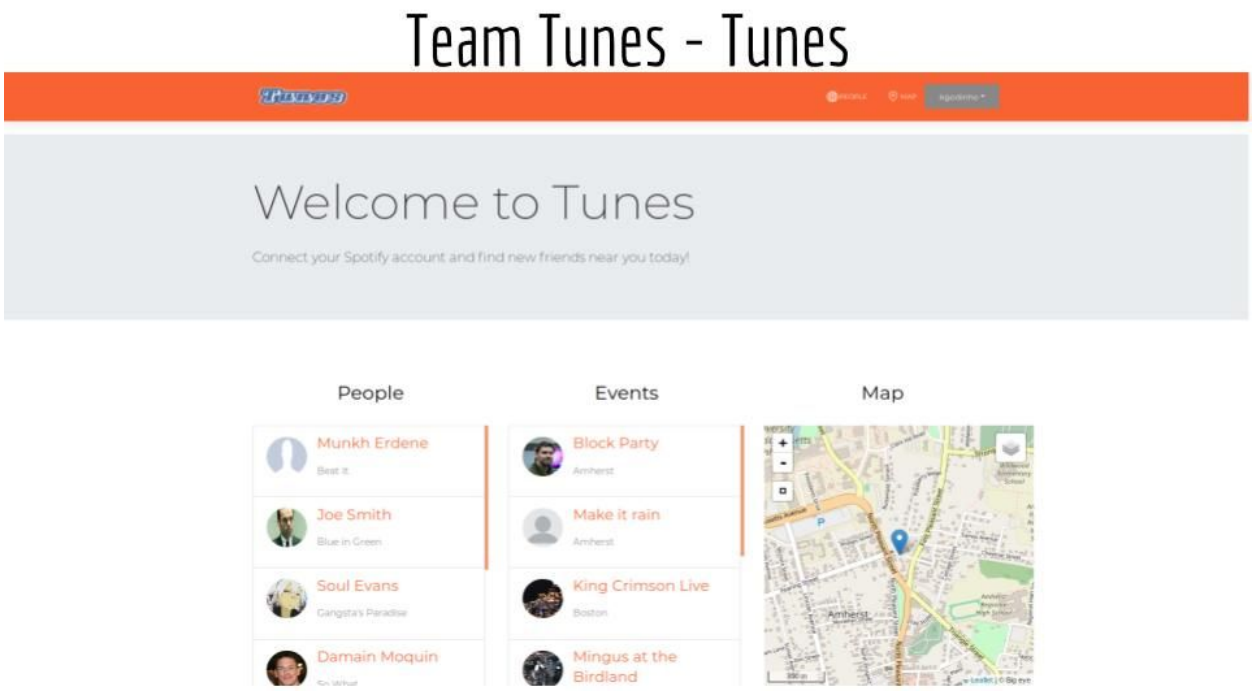
Our authentication process allows users to browse the site without currently being logged in, but in order to view their profile/settings pages or interact with events, they are prompted to either log in or create a profile. Upon becoming authorized by the site, they are able to view and edit their profile information, directly modifying the underlying database as a result. Similarly, users will be able to edit and create event listings on the site, as well as

browse and RSVP to them. We created the group “Event Admin” for users to give them the permission to create, edit, and delete event pages within the site.

Team Choice

For the team choice component of our project, we chose to use the django-leaflet library to support an interactive map to display registered users and events on our site. This involved familiarizing ourselves first with the leaflet library during the mock UI stage of our application, then implementing it within the Django framework using a third-party library. To do so, we added latitude and longitude elements to our Profile and Event classes within our model then used our views to populate the home page and map templates with information retrieved from our model about users/events and their geographic location.

Conclusion

Through this project, each member of our group got exposed to the end-to-end process of web development from initial prototyping and UI design, to back-end database design and server upkeep, to user authentication and final stages of development. We were each able to learn about the many pieces that go into making a fully functional and dynamically-generated site run, while still focusing on particular elements of the site that interested us most during the development phase. Communication was never a problem for us, but we did run into occasional issues with Git merge conflicts and database information being deleted when changes were pushed to the master branch. While a few of us had experience with HTML/CSS before starting the project, we all would have benefitted from more experience using Git/version control prior to starting in order to avoid such issues.

Single Slide