

SoundScape

*Personalize Your
Playlist*

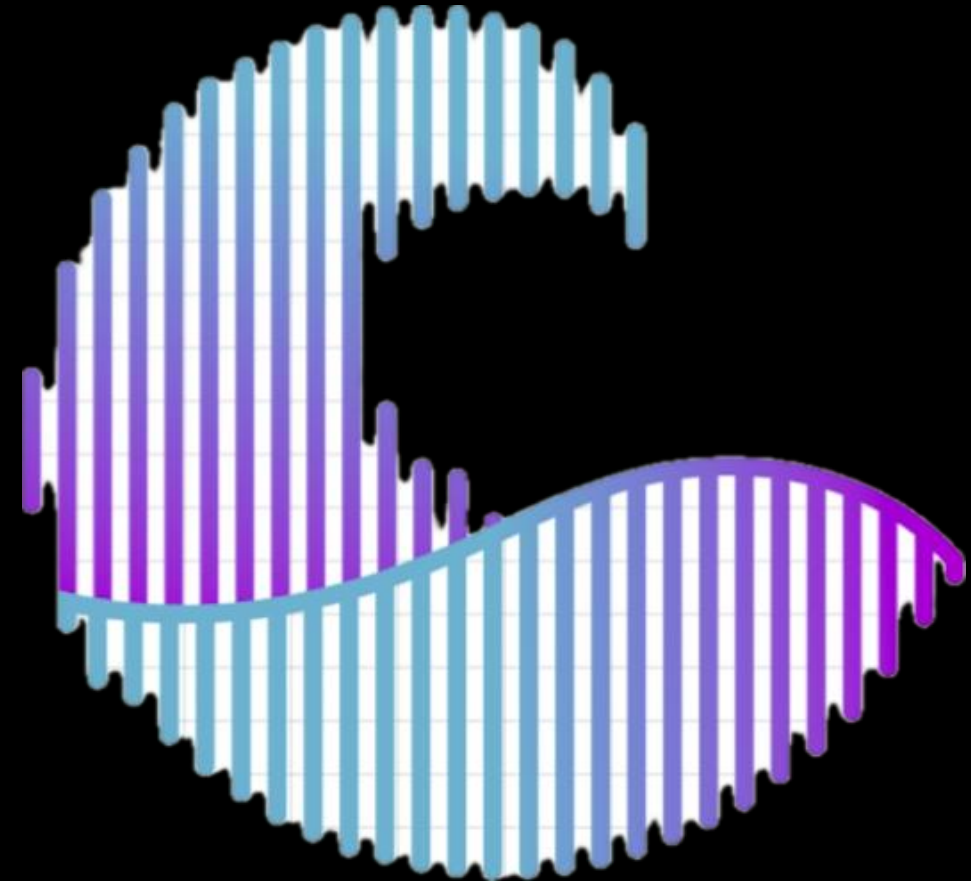
Group Members: Austin Schoster,
Derrick Mchale, Joe Schnizer

Advisor: William Hawkins



Project Description

- SoundScape is an AI-powered music app that generates personalized playlists by integrating with the Spotify and OpenAI APIs. The app tailors music recommendations to each user's preferences by analyzing various factors such as listening history, current mood, location, weather, time of day, and connected devices like cars or speakers. What's unique about this app is how personalized the playlists will be that it creates. It lets users control how much new music they want to discover, blending familiar tracks with fresh selections and accounting for the right mood or timing that may influence what someone wants to hear. This app will aim to create a seamless listening experience that evolves with the user's context and preferences
- Some useful project links
 - + GitHub: <https://github.com/Jschnizer/Senior-Design>
 - + Application: <http://localhost:3000>
 - NOTE: The app is still in development and therefore is not publicly available yet. For information on how to run the app, please see the user docs page [here](#) in our GitHub documentation



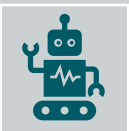
Project Goals



To develop an app that works with the user to provide personalized and accurate playlists based on various factors, ideally creating a playlist where the user will not want to skip any songs



To create an app that we would use regularly



Development of the group members' skills, particularly with training an AI model

Intellectual Merits



- Our project makes use of various APIs and data sources to gather data and produce personalized playlists
 - + Weather API
 - + Spotify API
 - + OpenAI PI
 - + User data/input
- The team researched developing a custom AI model to handle playlist creation
 - + No model exists that accounts for all the contextual inputs we are using to produce playlists
 - + We ultimately shifted to using the OpenAI API, but much was learned in the process
- Our team has learned a lot about front-end frameworks, UI development, and making an app that integrates a back-end and a front-end
 - + Server development
 - + React
 - + AI development

Broader Impacts

- We believe that this app will make creating personalized playlists more accessible to the average user, and that it will allow users to explore new music that they may like and have otherwise not been exposed to
- We think that users of our app will be able to easily create a playlist at any time that suits their current mood and situation
- We hope that our users will have their horizons expanded when it comes to what kinds of music they listen to



Design Specifications

- As it stands, our project is built on three main components
 - + Front-end UI built using React
 - + Back-end server running using Node.js and Express which forwards all necessary API requests to and from the front-end
 - + AI model running on a Python Flask server
- General flow
 - + The user logs into the app using their Spotify login
 - + The user tunes the parameters in the UI to customize their playlist and then clicks “generate playlist”. The app will then make various API requests to get all the necessary data and trigger the AI to make recommendations. All API requests currently go through the back-end and then out to the various 3rd parties and the AI model
 - + The user data is sent in the request to OpenAI and then it makes recommendations which are displayed to the user
 - + The user views these recommendations, decides which ones to add to their playlist, and exports the playlist to Spotify where they can listen to it

Design Diagrams

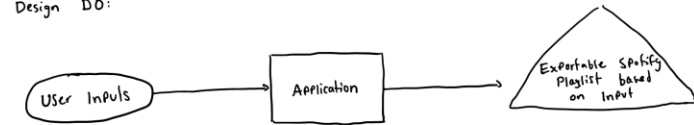
Wednesday, September 18, 2024 12:43 PM

SoundScape: A Spotify Companion App

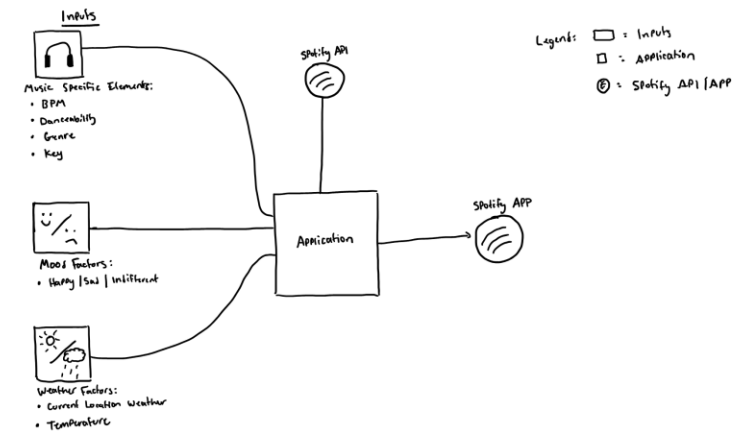
The goal of this project is to create a Spotify companion application that can create personalized playlists based on user given criteria and environmental factors.

Legend: ○ = User
□ = Application
△ = Result

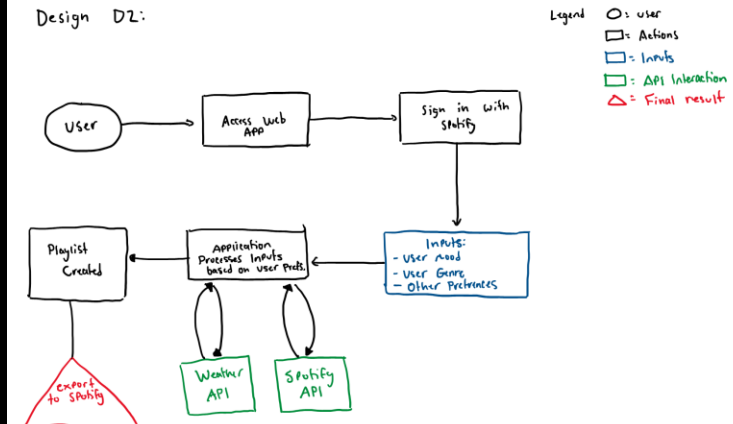
Design D0:



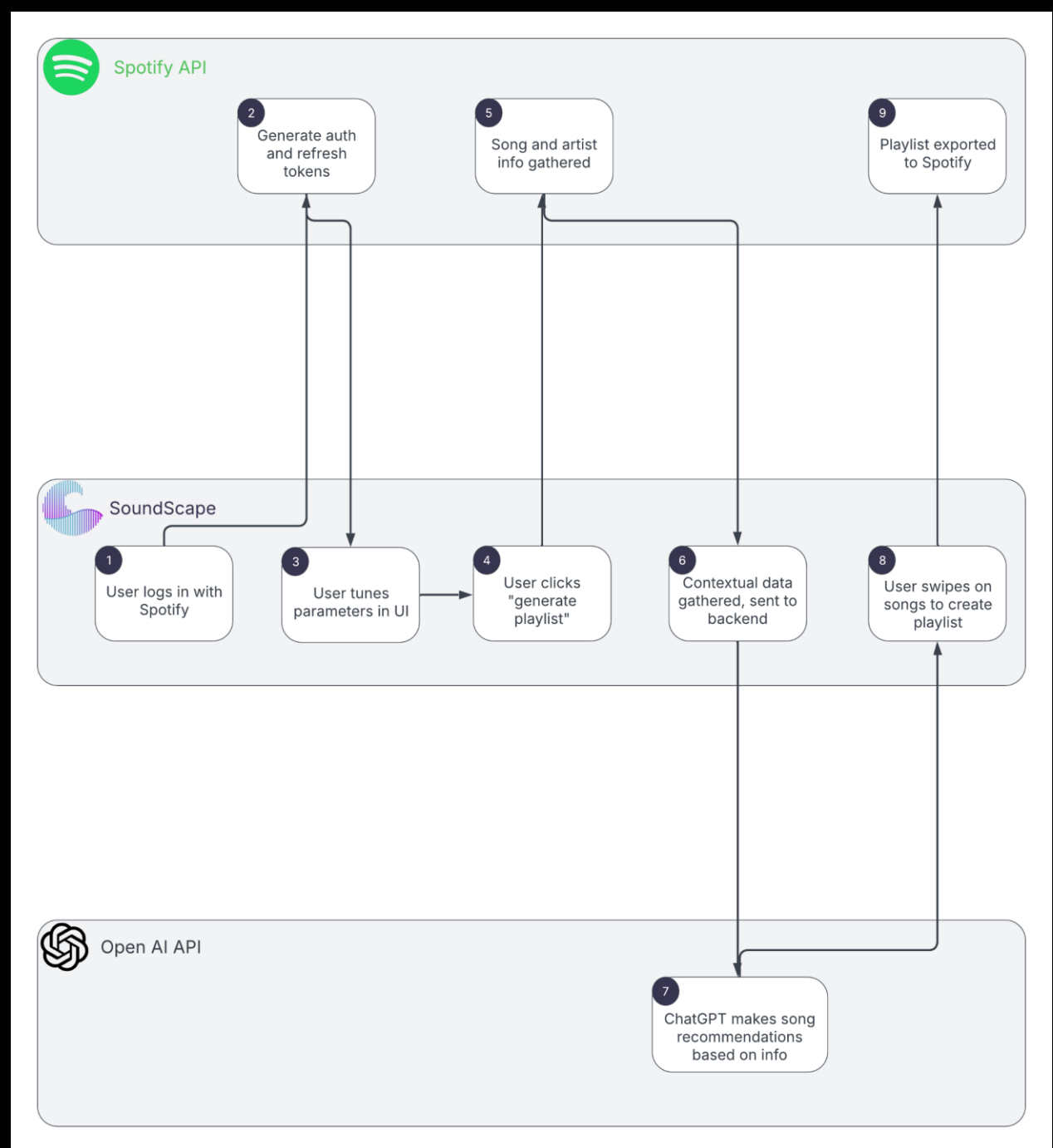
Design D1:



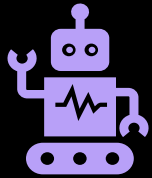
Design D2:



Flow Diagram



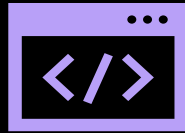
Technologies



AI

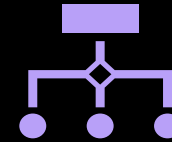
SoundScape uses OpenAI API calls as the AI system to make music recommendations

It was originally intended that a custom model would be developed, but many issues arose that could not be resolved



UI

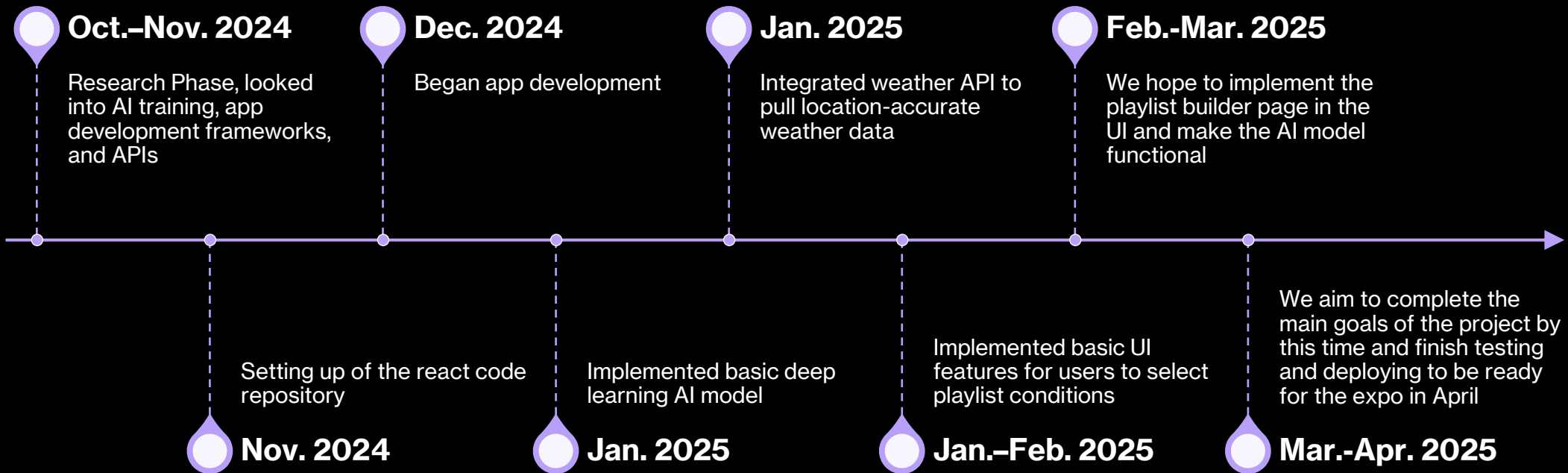
The front-end of SoundScape is developed using the JavaScript framework React. Some other libraries for specific UI components such as the cards were used



Back-End

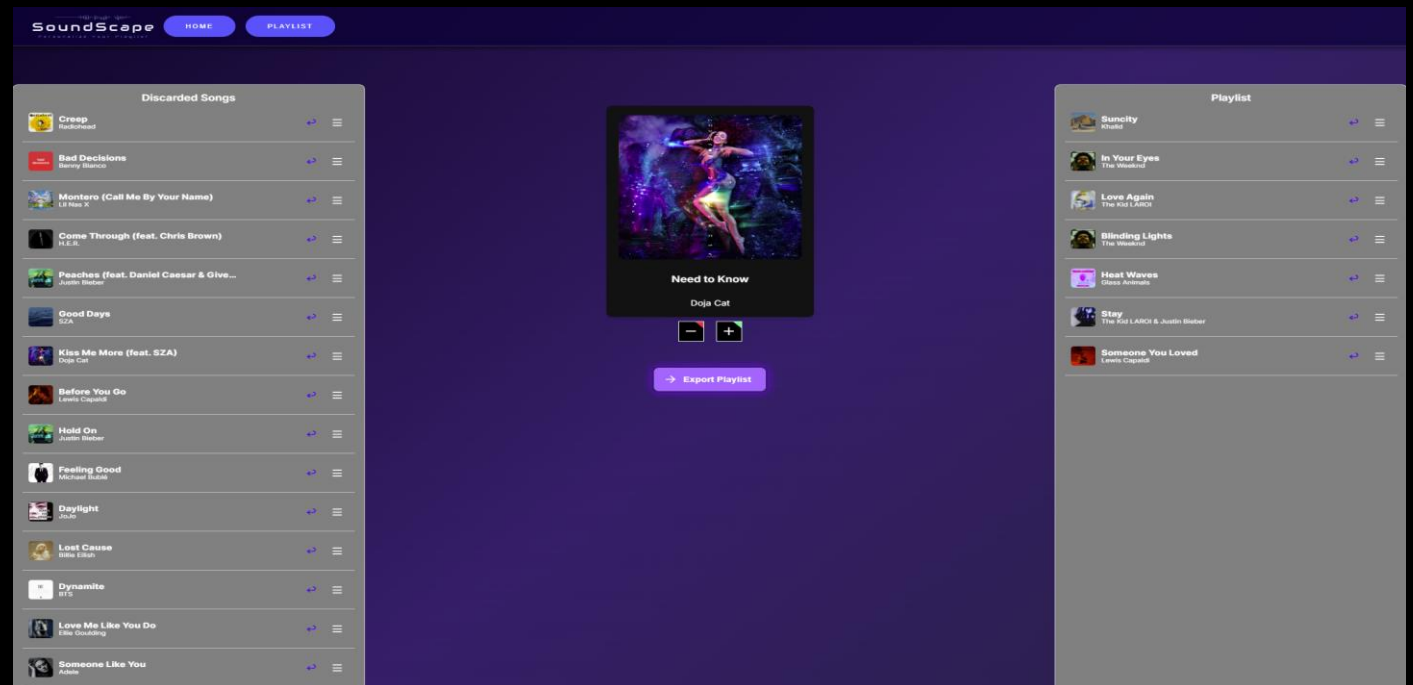
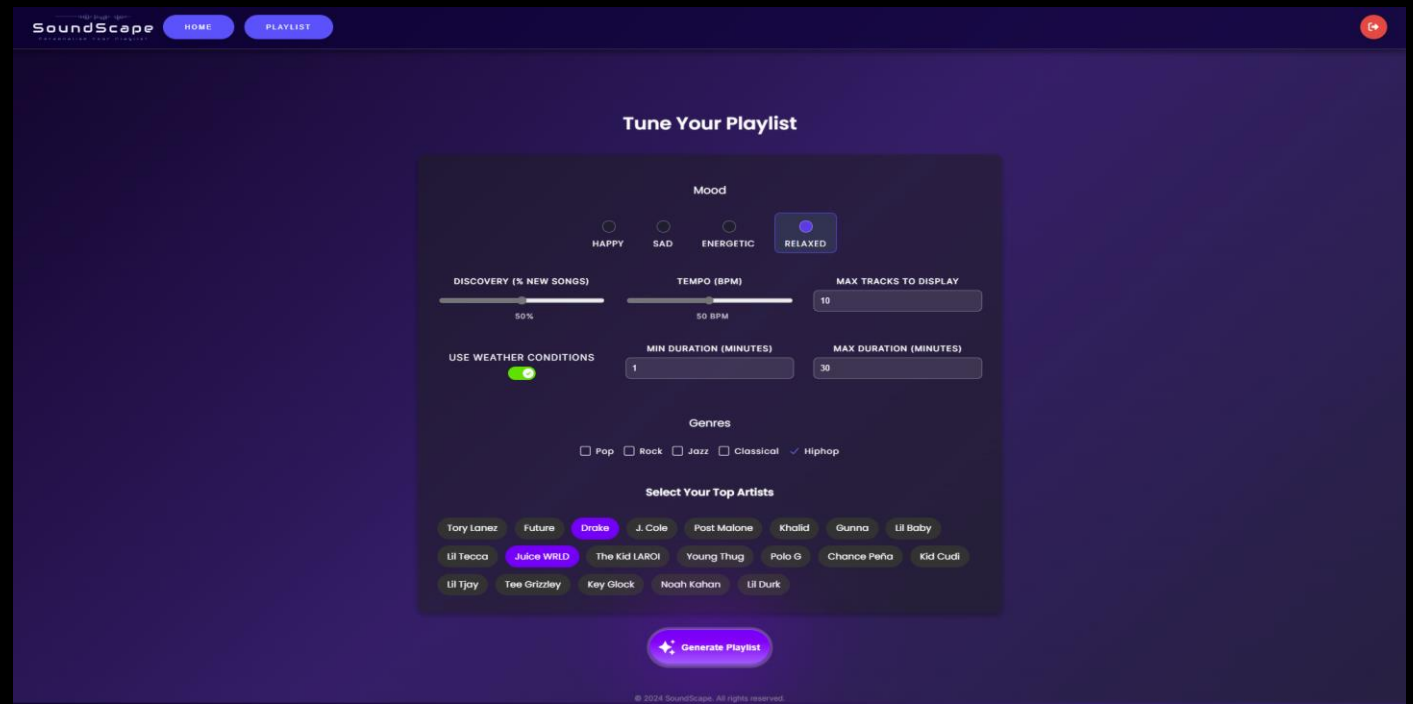
The back-end of the app uses various technologies including Node.js and Express to run the servers that make the API requests to get necessary data

Milestones



Results

- What we currently have done
 - + Webpage running on a server using Node.js and express
 - + Functioning UI that allows the user to select preferences for several variables, as well as accept or reject each suggested song using an intuitive swipe left or right mechanic
 - + API calls to OpenAI to handle the AI music recommendation system
 - + Validation system that ensures recommended songs exist in Spotify
 - + Pulling location-accurate weather data from a weather API
 - + Some images show the basic version of the app as it stands
- What still needs to be done
 - + Work on AI model – we need to refine the recommendation system so that it suggests better music based on the information provided
 - + Work on the UI – the playlists page and swiping functionality need to be implemented and the whole site needs to be refined
- Our final deliverable will be the fully functional app that will be publicly available for anyone to use at the expo



Challenges

A significant challenge that we faced was obtaining track data to feed to our AI model. Near the start of development, we discovered that Spotify had removed the ability to obtain users' listening data from their API, so we needed to find an alternative. After we switched to using the OpenAI API, this was no longer an issue

An additional challenge we faced was trying to develop and implement the AI model that would make music recommendations. We needed to figure out how to make an AI model and have it be available to use in our application. Unfortunately, although we were able to learn a lot when trying to do this, we ultimately decided to use the OpenAI API to handle our AI music recommendations instead of a custom model due to the challenges we faced

Another challenge we faced was creating the UI since none of the group members had much experience with React or UI work. We were able to utilize various resources to learn about React and create a user interface that works well for our app

We also had to navigate around our very busy work and school schedules during our senior years that made making time to develop the project difficult