SoundScape

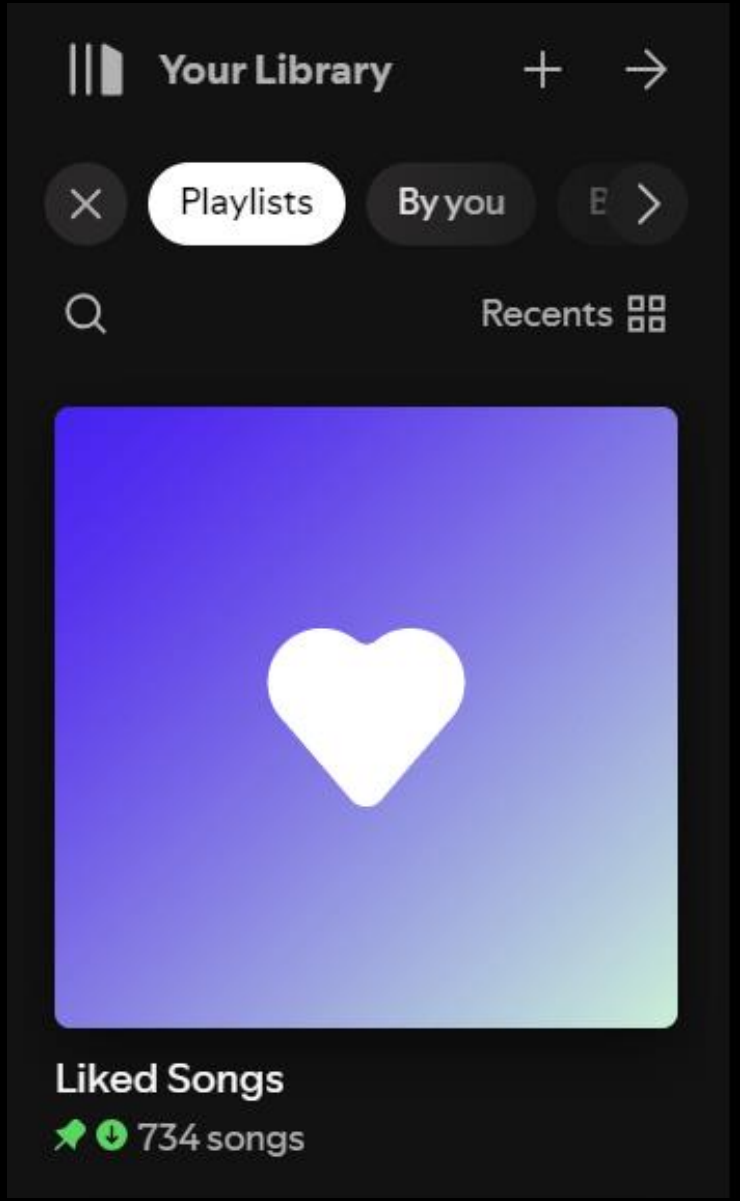
Personalize Your Playlist

Group Members: Austin Schoster, Derrick Mchale, Joe Schnizer

Advisor: William Hawkins

Project Description

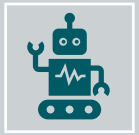
- SoundScape is an AI-powered music app that generates personalized playlists by integrating with the Spotify API. The app tailors music recommendations to each user's preferences by analyzing various factors such as listening history, current mood, location, weather, time of day, and connected devices 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

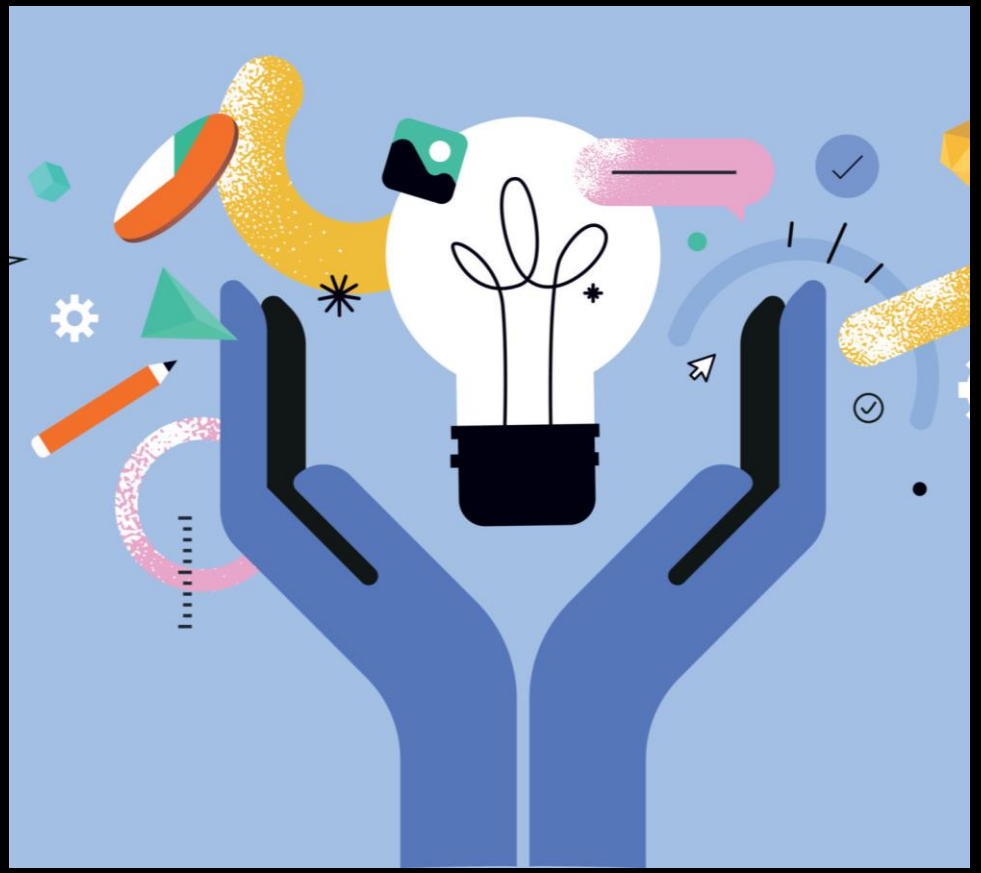
To develop an app that works with the user to provide personalized and accurate playlists based on various factors, ideally creating a playlist where the user will not want to skip any songs

To create an app that we would use regularly



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

Intellectual Merits



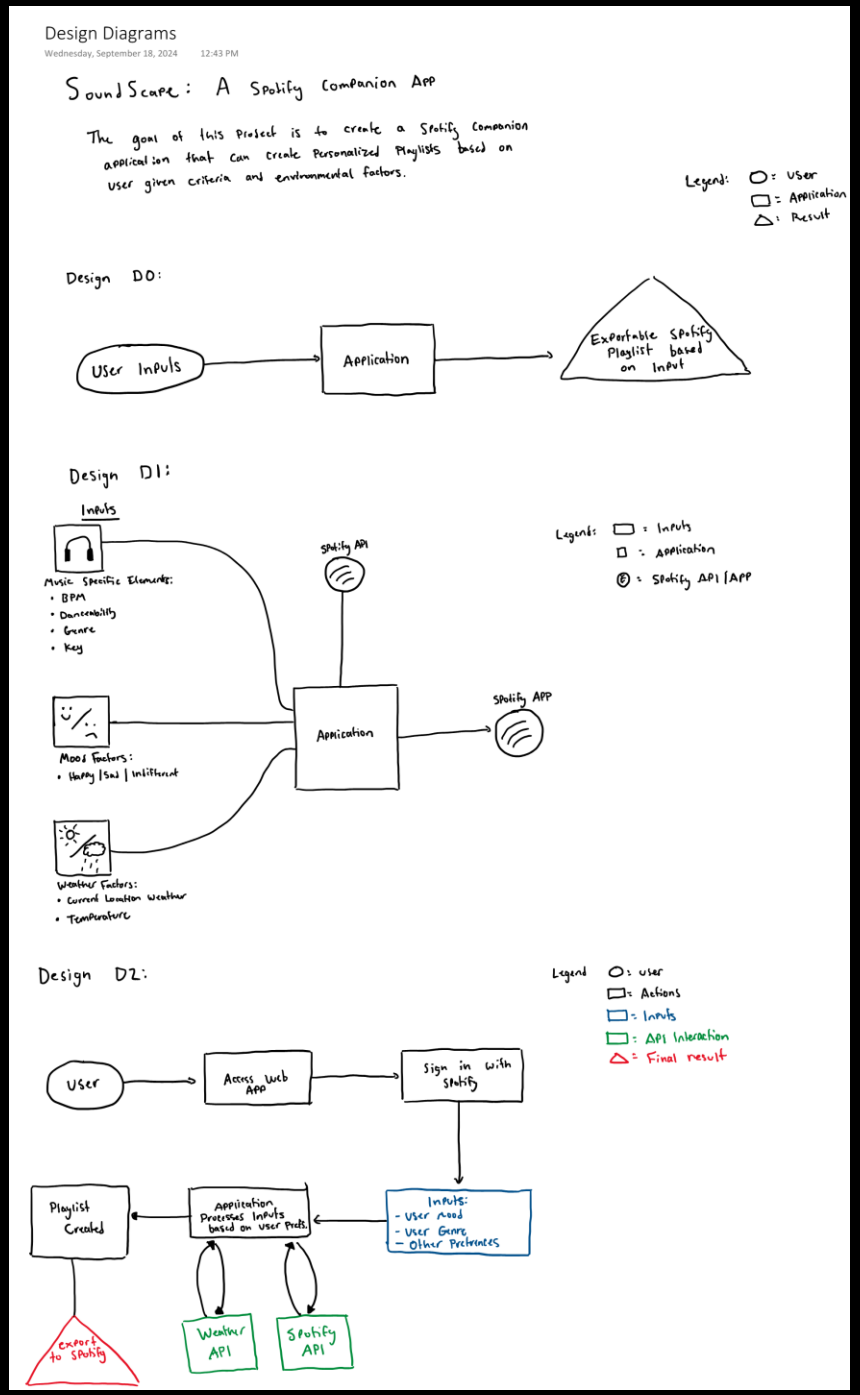
- Our project makes use of various APIs and data sources to gather data and produce personalized playlists
  - + Weather API
  - + Spotify API
  - + User data/input
- The team is developing a custom AI model to handle playlist creation
  - + No model exists that accounts for all the contextual inputs we are using to produce playlists
- 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

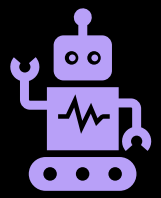
- We believe that this app will make creating personalized playlists more accessible to the average user, and that it will allow users to explore new music that they may like and have otherwise not been exposed to
- We think that users of our app will be able to easily create a playlist at any time that 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 triggers the AI to make suggestions. 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 AI model takes in the user data and 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



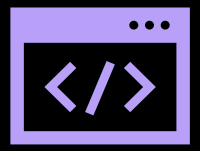
Technologies



AI

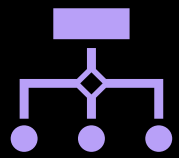
The AI technology developed and integrated into SoundScape is still very much in development

Currently using deep learning libraries (mainly TensorFlow and Keras) to implement a basic model



UI

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

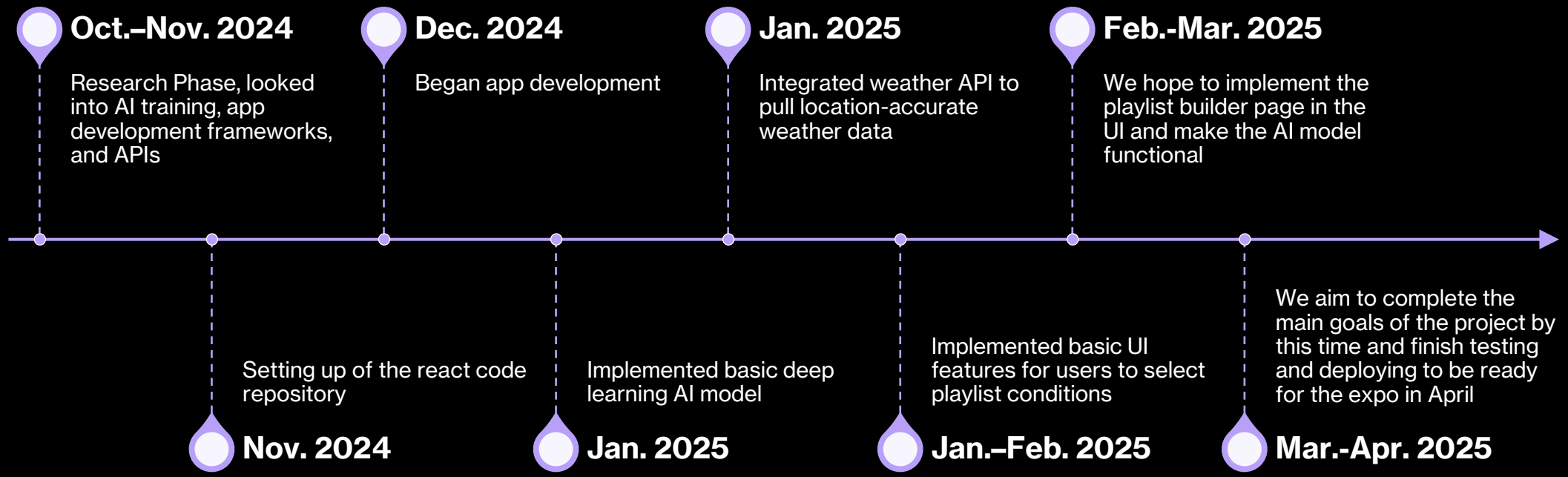


Back-End

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

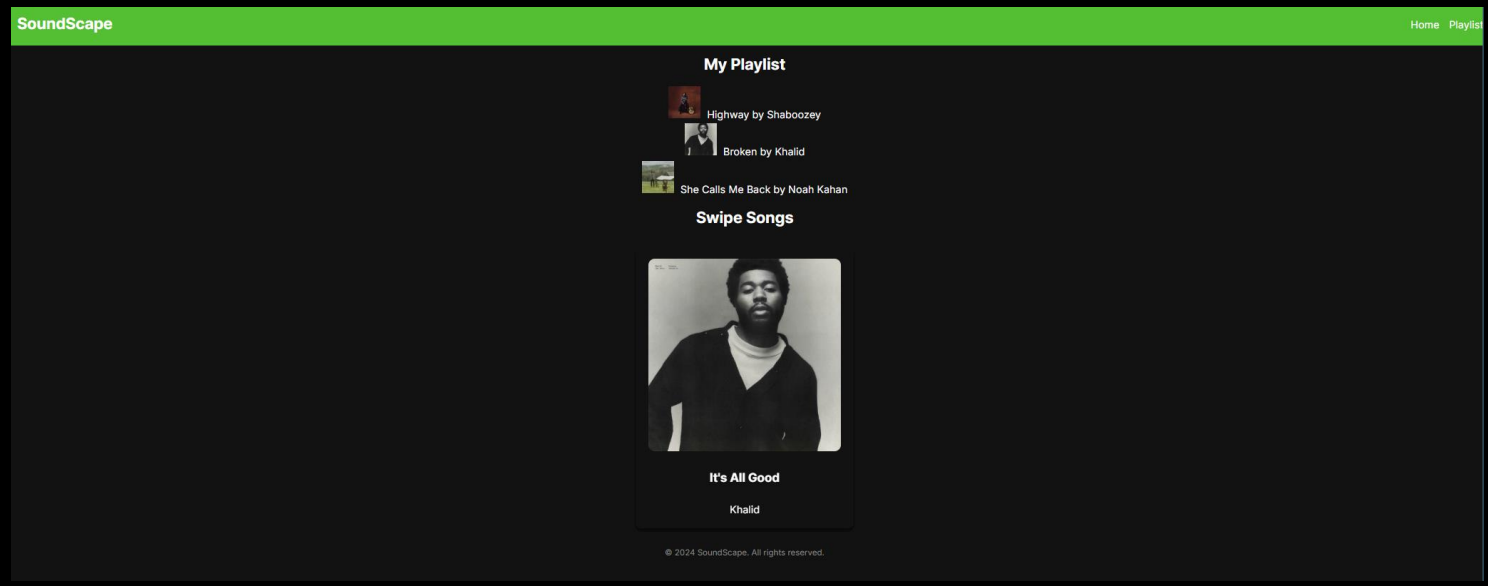
A Python Flask server is also running to handle requests made to the AI model at the moment

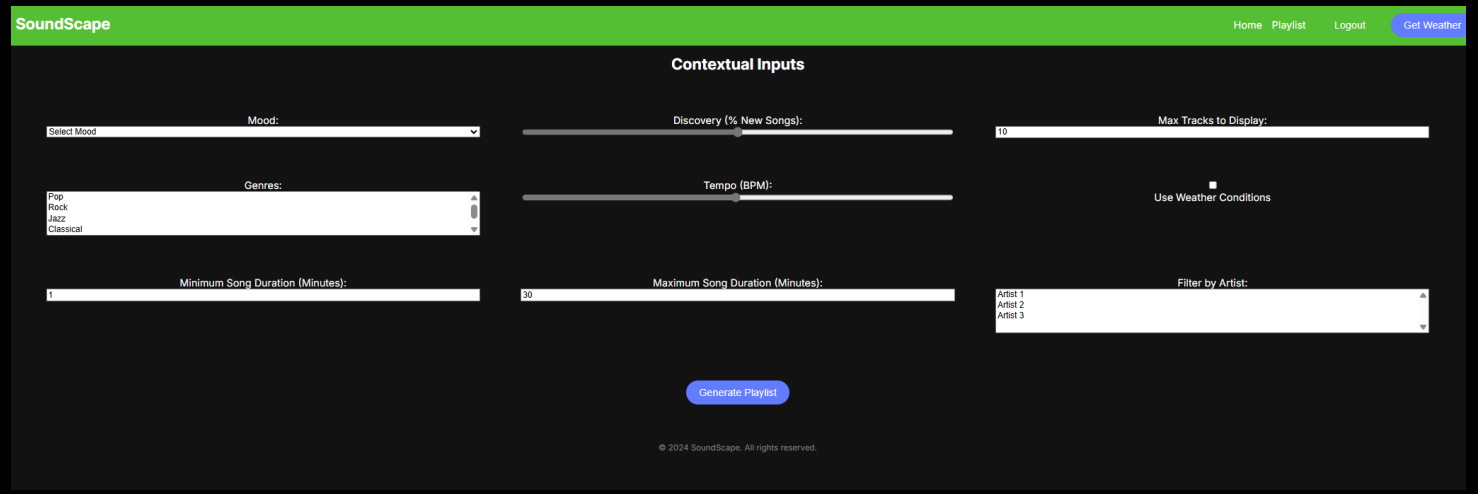
Milestones


Oct.–Nov. 2024
Research Phase, looked into AI training, app development frameworks, and APIs
Dec. 2024
Began app development
Jan. 2025
Integrated weather API to pull location-accurate weather data
Feb.-Mar. 2025
We hope to implement the playlist builder page in the UI and make the AI model functional
Setting up of the react code repository
Nov. 2024
Implemented basic deep learning AI model
Jan. 2025
Implemented basic UI features for users to select playlist conditions
Jan.–Feb. 2025
We aim to complete the main goals of the project by this time and finish testing and deploying to be ready for the expo in April
Mar.-Apr. 2025

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
  - + Basic AI model using deep learning libraries, which will be trained to suggest songs to the used based on the attributes selected in the UI
  - + 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 create training data and refine the model so it can actually make suggestions
  - + 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

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

Another challenge we have faced is with getting the “Tinder” cards implemented in the UI. We have encountered many errors trying to use pre-existing libraries, so we are currently using custom code to implement that