Spotify PULSE

Personalized User-Learned Social Experience

Sprint 2 Planning

Team 18
Rheaa Sharma
Noah Stern
Dane Shaffer
Alex Polivka
Jason Zheng
Collin “Bodhi” Scott

Sprint Overview:

In this second sprint, we hope to first offer the user more advanced recommendations based on adjustable parameters including danceability, tempo, and acousticness, as well as emotion-based presets. Advanced statistics will also be accessible from Spotify downloaded data allowing for much more data to be tracked and shown to the user. We aim to parse and present this data in a manner that is easy to comprehend, but allows for extra depth if the user wishes. We also aim to implement the PULSE AI chatbot with its complete functionality, as well as features such as themes, backgrounds, and more robust graphing tools to let users fully express themselves through the PULSE application.

Scrum Master: Noah Stern

Meeting Plan:

- Mondays 6:00 PM
- Wednesday 8:30 PM with Project Coordinator

Risks and Challenges:

Because team members were able to get accustomed to web development and the frameworks/languages associated with it in sprint one, the biggest challenge in sprint two will be the portions that the team had not worked with in the first sprint.

A significant hurdle is also the parsing of advanced statistics from Spotify. Our team has not yet received our advanced statistics from Spotify since it takes up to 30 days, which is another issue. However, from the preview Spotify provides, we know that information such as every song ever listened to will be included, and extracting meaningful data from said data dump and storing it will be a considerable challenge. Similarly, creating the PULSE chatbot, not to mention implementing it, is a semi-unknown quantity that will take significant research to achieve.

To overcome these challenges, our team aims to take our experiences from sprint one and make sprint two our most efficient and well-planned sprint yet. Having proper estimates which account for the time taken for research and integration would reduce our stress when working with unknown fields. Also, by following the same framework that we have created already for the website, starting tasks will become easier.

Current Sprint Details:

User Story #2c:

As a user, I would like to have custom backgrounds

#	Description	Time	Owner
1	Create UI for selecting background	1h	Alex
2	Implement custom background on pages	2h	Alex
3	Add preset backgrounds	1h	Alex
4	Create backend for background upload and background pulling	1h	Dane
5	Set up background storage system on the server	1h	Bodhi
6	Set up background file path storage in the database and ensure database and server are always synced	1h	Bodhi
7	Manual testing and testing documentation	1h	Alex

Acceptance Criteria:

1. Given the user is registered with PULSE, when the user selects a background, then applicable pages will change their background.
2. Given the user has a valid picture file, when the user wants to select a background, then they are given the option to add a custom image.
3. Given the user is registered with PULSE, when the user chooses a custom background, then the new background will be retained between logins.
4. Given the user is registered with PULSE, when the user wants to choose a background, then the user will have access to preset backgrounds.
5. Given the user has not set a custom background, when the user logs in, then the webpage will have the default background.

User Story #2d:

As a user, I would like to have preset themes to change up the interface's look

#	Description	Time	Owner

1	Create UI for selecting theme colors	2h	Alex
2	Create UI for managing custom themes	1h	Alex
3	Implement custom colors on pages	3h	Alex
4	Add preset theme colors	1h	Alex
5	Create backend for custom theme upload	1h	Dane
6	Store preset themes in the database	0.25h	Bodhi
7	Manual testing and testing documentation	1h	Alex

Acceptance Criteria:

1. Given the user is on the settings page, when the user inputs a custom theme, then applicable pages will change their colors.
2. Given the user is on the settings page, when the user chooses a custom theme, then the new theme will be retained between logins.
3. Given the user is on the settings page, when the user wants to choose a color theme, then the user will have access to preset color themes.
4. Given the user is on the settings page, when the user wants to create a custom color theme, then they can name and save it for future use.
5. Given the user has created custom themes, when the user wants to retrieve an old custom theme, then they can do so on the settings page.
6. Given the user has created custom themes, when the user wants to rename or delete an old custom theme, then they can do so on the settings page.

User Story #2e:

As a user, I would like to be able to view my Pulse friends

#	Description	Time	Owner
1	Create a storage method for friends list with packers/unpackers and getters.	1h	Bodhi
2	Create database functions for editing friends list through adding or removing friends.	0.25h	Bodhi
3	Create Database storage methods for friend requests with packers/unpackers and getters	0.25h	Bodhi

4	Create Database functions for editing friend requests	0.25h	Bodhi
5	Populate friends with their chosen icons	1h	Bodhi
6	Create UI for friends page that accommodates any amount of friends and has an add friend button, request button, and updates asynchronously between different accounts	2h	Bodhi
7	Create UI for friends tab on the right side of the screen that accommodates any amount of friends, shows any relevant information and lets you scroll	2h	Bodhi
8	Create an add friends tab on the friends page where the user can search for friends in a search bar and send friend requests	1h	Bodhi
9	Create a requests tab on the friends page where the user can see friend requests and accept or reject them	1h	Bodhi
10	Add functionality to add friends and pull list	1h	Dane
11	Add functionality to unadd friends	0.5h	Dane
12	Manual testing and testing documentation	1h	Bodhi

Acceptance Criteria:

1. Given that a user has no friends, when they view the friends tab on the right side of their screen, then it will be empty.
2. Given that a user has no friends, when they view the friends page, then it will say no friends and prompt them to send an invitation.
3. Given that a user wants to add friends, when they click to add friends on the friends page, then they are able to search for friends and send them invitations.
4. Given that a user has friend requests, when they enter the requests tab, they are able to view the requests and accept or decline them.
5. Given that a user wants to remove a friend, when they are in the friends page, then they are able to click on friends and remove them.

User Story #3b:

As a user, I would like to be able to export Spotify data and import it into PULSE to get the following statistics (“Advanced Statistics” using Spotify’s “Extended Streaming History”)

#	Description	Time	Owner
1	Parse through Spotify data text file and store it all locally	1h	Noah
2	Use this data to get the advanced statistics information needed and store it in a way that is both space-efficient and allows for easy accessing later	4.5h	Noah
3	Create and implement storage strategy + packer/unpacker for advanced statistics that will be stored in the database and delivered to the backend at later times.	3h	Bodhi
4	Create UI for uploading Spotify data	1h	Bodhi
5	Create page with information telling the user how they can get their extended streaming history from Spotify	1h	Bodhi
6	Add element to tell user if advanced stats has been downloaded & parsed by the server	0.5	Noah
7	Synchronize backend routes with frontend	0.5h	Noah
8	Call backend routes from frontend	0.5h	Bodhi
9	Automated unit testing for all advanced stats info with option to run it on a website route	2h	Noah

Acceptance Criteria:

1. Given the user is logged into PULSE, when the user wants access to their advanced statistics, then they can go to a page that gives them information on how to obtain this data.
2. Given the user has access to their extended streaming history data, when the user wants to give data to PULSE, then the user can upload data to PULSE.

3. Given the user has uploaded their extended streaming history to PULSE, when the user wants to check the status of data parsing, then the user can check the status of advanced data parsing.
4. Given the user has already uploaded their data to PULSE, when the user wants to give more data to PULSE, then the user can upload new data to PULSE
5. Given the user is logged into PULSE, when the user uploads incorrect bad data, then the user is alerted to the fact that the data is bad.

User Story #3c & #3d:

As a user, I would like for number of minutes listened (#3c) and percentage of total music listened (#3d) to be tracked in my listening activity (not using API, but using Advanced Statistics from extended user streaming history)

#	Description	Time	Owner
1	Extract appropriate portions of data from advanced statistics and ensure its passed to backend in a parsable manner	0.5h	Bodhi
2	Create compatibility for graphs to display this information	0.75h	Noah
3	Synchronize frontend and backend routes	1h	Noah
4	Manual testing and documentation	1h	Noah

Acceptance Criteria:

1. Given the user has uploaded their extended streaming history to PULSE, when the user logs in, then their data regarding number of minutes listened to Spotify over time and their percentage of total music listened to is tracked and available on the statistics page.
2. Given the user is on the statistics page, and they have uploaded their extended streaming history, then their number of minutes listened to Spotify over time is available in graph format.
3. Given the user is on the statistics page, and they have uploaded their extended streaming history, then their data regarding percentage of total music listened to by song is available in list format.
4. Given the user is on the statistics page, and they have not uploaded their extended streaming history, then the option to display their number of minutes listened to Spotify over time is unavailable.

5. Given the user is on the statistics page, and they have not uploaded their extended streaming history, then the option to display their percentage of total music listened to is unavailable.

User Story #3e:

As a user, I would like for times when music was listened to (morning, afternoon, night, etc.) in my listening activity (not using API, but using Advanced Statistics from extended user streaming history)

#	Description	Time	Owner
1	Extract appropriate portions of data from advanced statistics and ensure its passed to backend in a parsable manner	0.5h	Bodhi
2	Create compatibility for graphs to display this information	0.75h	Noah
3	Synchronize frontend and backend routes	1h	Noah
4	Manual testing and documentation	1h	Noah

Acceptance Criteria:

1. Given the user has uploaded their extended streaming history to PULSE, when the user logs in, then their data regarding times their music was listened to is tracked and available as a graph on the statistics page.
2. Given the user is on the statistics page, and they have uploaded their extended streaming history, when they select a timing graph, then their data regarding times their music is displayed.
3. Given the user is on the statistics page, and they have not uploaded their extended streaming history, then the option to display their data regarding times their music was listened to is unavailable.
4. Given the user is on the statistics page, when they have a timing graph, then the option to choose a time section (morning, afternoon, night, overall) is displayed.
5. Given uploads their extended streaming history a second time, when they navigate to the statistics page, then the graph automatically updates.

User Story #3f:

As a user, I would like for times when songs were listened to fully or skipped or repeated in my listening activity (not using API, but using Advanced Statistics from extended user streaming history)

#	Description	Time	Owner
1	Extract appropriate portions of data from advanced statistics and ensure its passed to backend in a parsable manner	0.5h	Bodhi
2	Create compatibility for graphs to display this information	0.75h	Noah
3	Synchronize frontend and backend routes	1h	Noah
4	Manual testing and documentation	1h	Noah

Acceptance Criteria:

1. Given the user has uploaded their extended streaming history to PULSE, when the user logs in, then their data regarding times songs were skipped is tracked and available on the statistics page.
2. Given the user is on the statistics page, and they have uploaded their extended streaming history, then their data regarding times songs were skipped is available in a calendar graph.
3. Given the user is on the statistics page, and they have uploaded their extended streaming history, then their data regarding times songs were skipped is available in a bar graph.
4. Given the user is on the statistics page, and they have uploaded their extended streaming history, then their data regarding times songs were skipped is available in a line graph.
5. Given the user is on the statistics page, and they have not uploaded their extended streaming history, then the option to display their data regarding times songs were skipped is unavailable.

User Story #3g:

As a user, I would like to see songs, artists, playlists, genres and favorite era of music sorted by the metrics above (not using API, but using Advanced Statistics from extended user streaming history)

#	Description	Time	Owner

1	Extract appropriate portions of data from advanced statistics and ensure its passed to backend in a parsable manner	0.5h	Bodhi
2	Create compatibility for graphs to display this information	0.75h	Noah
3	Synchronize frontend and backend routes	1h	Noah
4	Create frontend user selection for favorites criteria	1h	Bodhi
5	Store user choice for what counts as a favorite in database	0.25h	Bodhi
6	Handle backend logic that determines what data gets passed to the frontend based on what qualifies as a user favorite for songs, artists, playlists, genres, era of music	1h	Noah
7	Manual testing and documentation	1h	Noah

Acceptance Criteria:

1. Given the user has uploaded their extended streaming history to PULSE, when the user logs in, their data regarding their favorites by number of minutes listened to are tracked and available on the statistics page.
2. Given the user is on the statistics page, and they have uploaded their extended streaming history, their favorite songs by number of minutes listened to is available in graph format.
3. Given the user is on the statistics page, and they have uploaded their extended streaming history, their favorite artists by number of minutes listened to is available in graph format.
4. Given the user is on the statistics page, and they have uploaded their extended streaming history, their favorite playlists by number of minutes listened to is available in graph format.
5. Given the user is on the statistics page, and they have uploaded their extended streaming history, their favorite genres by number of minutes listened to is available in graph format.
6. Given the user is on the statistics page, and they have uploaded their extended streaming history, their favorite era of music by number of minutes listened to is available in graph format.

7. Given the user is on the statistics page, and they have not uploaded their extended streaming history, the option to display their data regarding their favorites by minutes listened to is unavailable.
8. Given the user has uploaded their extended streaming history, and they change their criteria for what counts as a favorite, their advanced statistics regarding their favorites are updated to reflect this once the page is refreshed.

User Story #3p:

As a user, I would like to be able to create custom graphs with specified data and formats

#	Description	Time	Owner
1	Modify frontend/backend to allow for legend toggle when creating graph	2h	Jason
2	Modify 3 existing graphs with their respective settings for legend display	1.5h	Jason
3	Add toggle to allow users to select play music or pull up Spotify page	1.5h	Jason
4	Modify structure to allow for multiple data on graphs	2h	Jason
5	Add additional themes to display for graphs	1h	Jason
6	Add tooltip on title of graph to display graph settings	1h	Jason
7	Add option for custom axis titles w/ toggle & only for valid graph types	2h	Jason
8	Modify popup validation to go from “graph” to “data” instead of the reverse	0.25h	Jason
9	Add validation for advanced stats	0.5h	Jason
10	Add option to display text data instead of graphs	1.5h	Jason
11	Add error-handling and all validation required to create a graph w/ testing	2h	Jason

Acceptance Criteria:

1. Given user has opened “Add Graph” menu, when a graph that can have a legend is selected, then allow that graph to toggle legend on and off
2. Given user has opened “Add Graph” menu, when a graph that can have “play music” or “Spotify link” option, then allow that graph to toggle between “play music” or “Spotify link”
3. Given user has opened “Add Graph” menu, when a graph that can have multiple data displayed, then allow user to select multiple data variables
4. Given user has created a graph, when the user hovers over a graph title, then data such as data type and settings are displayed
5. Given user has opened “Add Graph” menu, when a graph that can have axis titles is selected, then allow user to input axis titles
6. Given user has opened “Add Graph” menu, when a graph that can have data displayed in text, then allow user to display data in text

User Story #3q:

As a user, I would like to be able to have many choices of graphs to display data

#	Description	Time	Owner
1	Create Bump graphs for top songs/artists data	2h	Jason
2	Add calendar graph for data with timestamps	2h	Jason
3	Add radial bar graph for bar graph data	2h	Jason
4	Modify existing bar graph to have a horizontal bar graph equivalent	1h	Jason
5	Add scatter plot for line graph data types	2h	Jason
6	Test creation w/ all variations of inputs for graphs	1h	Jason

Acceptance Criteria:

1. Given user has opened “Add Graph” menu, when graph type called “Bump Graph” is selected, user can create a bump graph
2. Given user has opened “Add Graph” menu, when graph type called “Calendar Graph” is selected, user can create a calendar graph
3. Given user has opened “Add Graph” menu, when graph type called “Radial Bar Graph” is selected, user can create a radial bar graph

4. Given user has opened “Add Graph” menu, when graph type called “Horizontal Bar Graph” is selected, user can create a horizontal bar graph
5. Given user has opened “Add Graph” menu, when graph type called “Scatterplot” is selected, user can create a scatter plot

User Story #3r:

As a user, I would like to be able to view my friends’ statistics as per previously mentioned user tracked statistics

#	Description	Time	Owner
1	Create backend functions to pull friend statistics from PULSE	2h	Dane
3	Modify frontend graphs to store whose data the graph belongs to	2h	Jason
4	Modify frontend graphs to allow multi-data graphs to also display multi-user data	2h	Jason
5	Test most combinations of graphs w/ friends data	1.5h	Jason

Acceptance Criteria:

1. Given that the user has opened an “Add Graph” menu, they see a prompt to add a friend.
2. Given that the user has unfriended a user, when the user tries to display a graph with that friend’s data, then that data is not displayed
3. Given that the user has no friends, when the user tries to add a graph, they will not be able to add friend data to a graph
4. Given that the user has opened “Add Graph” menu, when a graph that can have multiple data, user can add friend data to display as well
5. Given that the user has added a friend to a graph, when the graph is displayed, then the friend's name is displayed.

User Story #3u & #3v:

As a user, I would like to see advanced data percentage-wise including what genres I’ve listened to (#3u) and what emotions I experience when listening to Spotify (#3v) (not using API, but using Advanced Statistics from extended user streaming history)

#	Description	Time	Owner

1	Extract appropriate portions of data from advanced statistics and ensure its passed to backend in a parsable manner	0.5h	Bodhi
2	Create backend to read emotions from Spotify list	2h	Dane
3	Create radar graph for displaying emotion data, and compatibility for statistics page to display emotion data	2h	Noah
4	Synchronize frontend and backend routes	1h	Noah
5	Manual testing and testing documentation	1h	Noah

Acceptance Criteria:

1. Given the user has uploaded their extended streaming history to PULSE, when the user logs in, their data regarding what genres they listened to are tracked and available on the statistics page.
2. Given the user is on the statistics page, and they have uploaded their extended streaming history, their data regarding what genres they listened to is available in graph format.
3. Given the user is on the statistics page, and they have uploaded their extended streaming history, then their data regarding what emotions they experience while listening to Spotify is available in graph format.
4. Given the user is on the statistics page, and they have not uploaded their extended streaming history, the option to display their data regarding what genres they listened to is unavailable.
5. Given the user is on the statistics page, and they have not uploaded their extended streaming history, then the option to display their data regarding what emotions they experience while listening to Spotify is unavailable.

User Story #3w:

As a user, I would like to see what emotions a song I select matches

#	Description	Time	Owner
1	Create backend function to pull current song & give emotions back	3h	Dane
2	Create compatibility for graphs to display this information	0.75h	Noah

3	Synchronize frontend and backend routes	1h	Noah
4	Manual testing and testing documentation	1h	Noah

Acceptance Criteria:

1. Given the user is on the statistics page, when they select to add a graph, then the song emotion analyzer graph is presented to them.
2. Given the user is on the statistics page and has a song analyzer, when they view the song analyzer, then a search bar to select a song to analyze is displayed.
3. Given the user is on the statistics page, when they select a song to analyze, then the song analyzer displays how well the songs match each emotion in a percentage based format.
4. Given the user is on the statistics page and has a song already being analyzed, when they change songs using the search box, then the graph percentage updates to the new song.
5. Given the user leaves the statistics page with an analyzer graph, when they return, then the last searched song is saved in the graph.

User Story #4a & #4b & #4e:

As a user, I would like for recommendations for songs to add on the playlist based on genres (#4a) & artists (#4b) & albums (#4e)

#	Description	Time	Owner
1	Develop front end navigation to a playlist page and create playlist page UI	1.5h	Bodhi
2	Add search bar for playlists	0.5h	Bodhi
3	Add display for recommendations with method for adding to playlist	1.5h	Bodhi
4	Develop backend for search bar for playlists backend	0.5h	Dane
5	Develop backend for playlist iterating over songs inside a playlist	2h	Dane
6	Develop recommendation finder for playlists based of genre	1h	Dane

7	Add dropdown to choose what recommendation should be based off of	0.5h	Alex
8	Develop backend for playlist recommendations based off of artists	0.5h	Dane
9	Develop backend for playlist recommendations based off of albums	1h	Alex
10	Manual testing and testing documentation	2h	Alex

Acceptance Criteria:

1. Given that a user wants to add songs to their playlist based on a specific criteria, when they navigate to the DJ mixer page and click on the playlist page, then they are able to see an option for adding songs to the playlist.
2. Given that the user wants to add songs to a specific playlist based on a specific criteria, when they search for the playlist, then they are able to select it as a target for the songs to be added to.
3. Given that a user wants to add songs to their playlist based on genre, when they are on the playlist page, then there is a dropdown that allows them to choose to add songs to the chosen playlist based on genre
4. Given that the user chooses to add songs to a specific playlist based on genre, when they view the song recommendations, then they are similar in genre to the songs already in the playlist.
5. Given that a user wants to add songs to their playlist based on the artists, when they are on the playlist page, then there is a dropdown that allows them to choose to add songs to the chosen playlist based on artist
6. Given that the user chooses to add songs to a specific playlist based on genre, when they view the song recommendations, then they are similar in artists to the songs already in the playlist.
7. Given that a user wants to add songs to their playlist based on the album, when they are on the playlist page, then there is a dropdown that allows them to choose to add songs to the chosen playlist based on artist
8. Given that the user chooses to add songs to a specific playlist based on the album, when they view the song recommendations, then they are similar in album choice to the songs already in the playlist.

User Story #4f:

As a user, I would like for recommendations for songs or playlists based on User emotion presets (informed by Spotify API)

#	Description	Time	Owner
1	Create UI for choosing emotion	1h	Alex
2	Create emotion presets for parameters	2h	Dane
3	Update parameters to display emotion presets	0.5h	Alex
3	Create UI to display song recommendations	2h	Alex
4	Retrieve recommendations using parameters	1h	Dane
5	Manual testing and testing documentation	1h	Alex

Acceptance Criteria:

1. Given the user is logged in, and the user navigates to the recommendation page, they are prompted with an emotion recommendation prompt.
2. Given the user is on the recommendation page, they can select an emotion to get recommendations from.
3. Given a user chooses an emotion, the parameter sliders update with the emotion target values.
4. Given the user is on the recommendation page, when they have an emotion selected and ask for recommendations based on the parameters, songs populate the recommendation area.
5. Given the user has recommendations displaying, when they select a new emotion and request recommendations, the songs are replaced by the new recommendations.

User Story #4h & #5g:

As a user, I would like a song and playlist activity monitor that displays the amount of times a song (#5g) or playlist (#4h) was played in an user-friendly manner on the playlist page (not using API, but using Advanced Statistics from extended user streaming history)

#	Description	Time	Owner
1	Extract appropriate portions of data from advanced statistics and ensure its passed to backend in a	0.5h	Bodhi

	parsable manner		
2	Create compatibility for graphs on statistics page to display song and playlist activity	0.75h	Noah
3	Create UI on the playlist page to display playlist activity	0.5h	Bodhi
4	Create UI on the song page to display song activity	0.5h	Bodhi
5	Synchronize frontend and backend routes	1h	Noah
6	Manual testing and documentation	1h	Noah

Acceptance Criteria:

1. Given the user has uploaded their extended streaming history to PULSE, when the user logs in, then their data regarding times songs and playlists were played is tracked and available on the playlist page.
2. Given the user is on the playlist page, and they have uploaded their extended streaming history, then their data regarding times playlists were played is available.
3. Given the user is on the song page, and they have uploaded their extended streaming history, then their data regarding times songs were played is available.
4. Given the user is on the statistics page, and they have uploaded their extended streaming history, then their data regarding times songs and playlists were played is available.
5. Given the user is on the statistics page, and they have not uploaded their extended streaming history, then the option to display their data regarding times songs and playlists were played is unavailable.

User Story #4k:

As a user, I would like to be able to set my emotion parameters with user chosen playlist

#	Description	Time	Owner
1	Create UI to choose playlist and create parameter set	2h	Alex
2	Create UI to manage custom parameter sets	1h	Alex
3	Backend for custom parameter set storage and	2h	Alex

	pulling parameters from playlist		
4	Store custom parameter set in database and create relevant retrieval and processing functions	1.5h	Bodhi
5	Manual testing and testing documentation	1h	Alex

Acceptance Criteria:

1. Given a user has created custom emotion parameter sets, when they go to the recommendation page, then they can rename and delete their custom emotion parameter sets.
2. Given a user is logged in, when they navigate to the recommendation page, then they see a prompt to create or edit a custom emotion parameter set.
3. Given a user selects to update their emotion preferences, they are presented with a prompt to choose one of their owned playlists.
4. Given a user is on the playlist page, when they select a playlist, then they are prompted with the ability to use that playlist to edit or create a user emotion parameter set.
5. Given a user selects an owned playlist, their personal emotion preferences are updated and can be viewed on the parameter sliders.

User Story #4I:

As a User, I would like to manage playlists through Pulse.

#	Description	Time	Owner
1	Add navigation to playlist management page	0.5h	Alex
2	Add track adding and frontend	1h	Dane
3	Add image management and frontend	2h	Dane
4	Add playlist creation	1h	Dane
5	Add track remover/ and track replacer	1h	Dane
6	Add track reorder	1h	Dane
7	Add playlist unfollow and follow	1h	Dane
8	Manual testing and testing documentation	1h	Alex

Acceptance Criteria:

1. Given a user is logged into Pulse, when they navigate to the dj mixer page, then they can see a prompt to navigate to the playlist management page.
2. Given a user is on the dj mixer page, when they navigate to the playlist management page, then they see a prompt to add a track.
3. Given a user is on the dj mixer page, when they navigate to the playlist management page, then they see a prompt to change playlist image.
4. Given a user is on the dj mixer page, when they navigate to the playlist management page, then they see a prompt to create playlists.
5. Given a user is on the dj mixer page, when they navigate to the playlist management page, then they see a prompt to remove tracks.
6. Given a user is on the dj mixer page, when they navigate to the playlist management page, then they see a prompt to reorder tracks.
7. Given a user is on the dj mixer page, when they navigate to the playlist management page, then they see a prompt to unfollow or follow playlists.

User Story #4m:

As a user, I would like to auto-generate playlists through Pulse based on genre

#	Description	Time	Owner
1	Add navigation to playlist generation page	0.5h	Alex
2	Create UI for playlist generation page	1h	Alex
3	Add playlist creation and auto-population	2h	Dane
4	Manual testing and testing documentation	1h	Alex

Acceptance Criteria:

1. Given a user is logged in, they can navigate to the playlist generation page.
2. Given a user is on the playlist generation page, they see a prompt to generate a playlist.
3. Given a user is on the playlist generation page, they can input a name for the newly generated playlist to have.
4. Given a user is on the playlist generation page, they can input if the newly generated playlist should be public.
5. Given a user is on the playlist generation page, they can input if the newly generated playlist should be collaborative.
6. Given a user is on the playlist generation page, they can select a genre to generate a playlist off of.

User Story #5b:

As a user, I would like to be able to find similar songs by specifying Spotify API target parameters such as loudness to a chosen song.

#	Description	Time	Owner
1	Add sliders to specify parameters based on Spotify's major parameters to the recommendation page.	1h	Bodhi
2	Add backend functionality to parameter setting	1h	Dane
3	Add functionality to store saved preferences in the database to use on future logins	0.5h	Bodhi
4	Add functionality to grab recommendations using user chosen parameters	2h	Dane
5	Populate page with recommended songs according to the given parameters	0.5h	Bodhi
6	Manual testing and testing documentation	1h	Alex

Acceptance Criteria:

1. Given a user is on the recommendation page, when they view the page then there are slider prompts for each major Spotify parameter.
2. Given a user is on the recommendation page, when they view the page then there is an option to select presets and create new presets to use for parameters.
3. Given a user is on the recommendation page, when they view the presets then there are automatically created presets based off of emotions to select from in the preset dropdown.
4. Given a user is on the recommendation page, when they have selected a preset then there is a prompt to receive recommendations based on the currently selected preset.
5. If a user chooses to receive recommendations, when they click the given button to receive recommendations, then the recommendations box auto-populates with recommended songs based on said parameters.

User Story #6a:

As a user, I would like to play games such as Guess who listens to this song as a quiz format and heads-up format

#	Description	Time	Owner
1	Create UI for guess who listen to this song	0.5h	Rheaa
2	Find a way to merge playlists between friends (assuming friend tracking is done in the database)	2h	Rheaa
4	Editing the UI for the heads-up format	1h	Rheaa
3	Create scoring tracking	0.5h	Rheaa
4	Send scores to backend to be stored in database	1h	Rheaa
5	Add the scores of previous games to the scores history on the main games page	1h	Rheaa
6	Manual testing and testing documentation	1h	Noah

Acceptance Criteria:

1. Given the user has chosen the quiz format, when the user starts the game, the game should play a random song. The user should be able to listen to the song and then provide their guess regarding who listens to the song.
2. Given the user has chosen the heads-up format, when the user starts the game, the game should play a random song but the skips should be handled differently from the quiz format.
3. Given the user has completed a game session, when the game session ends, the user's score should be displayed.
4. Given the user has played multiple game sessions, when the user navigates to the scores history page from the main games page, the scores history should be presented in a clear and organized manner.
5. Given the user has past game sessions, when the user in on the main games page, they should be able to see past games scores.

User Story #11a:

As a user, I would like to interact with a PULSE chatbot through a textbox

#	Description	Time	Owner
1	Create the UI for the ChatBot	2h	Rheaa
2	Create functionality to have the send button to	1.5h	Rheaa

	trigger the ChatGPT API		
3	On click of the Button call the ChatGPT API	0.2h	Rheaa
4	Time spent training the ChatGPT API to know specifics about our website	2h	Rheaa
5	Manual testing and documentation	1h	Noah

Acceptance Criteria:

1. Given the user has accessed the chatbot interface, when the user interacts with the UI, the chat interface should match the theme of the website.
2. Given the user has initiated a chat session, when the user clicks the send button to submit a message, the chatbot's response should be shown to the user.
3. Given the user has initiated a chat session, when the user clicks the send button to submit a message, the chatbot's response should be in a friendly manner.
4. Given the user is logged in, when the user navigates to the homepage, then the chatbot symbol should be shown to be accessed.
5. Given the user is logged in, when the user tries to interact with other elements on the page, then the chatbot button should not interfere with other elements' operation.

User Story #11b:

As a user, I would like to be able to get song recommendations per Genre, Artist, Location, Album and User emotion through PULSE chatbot

#	Description	Time	Owner
1	Setting up prompts for recommendations per Genre, Artist, Location,Album and User emotion through PULSE chatbot	2h	Rheaa
2	Testing and modifying prompts and setting them as the default prompts (for better user experience)	1h	Rheaa
3	Create functionality to send required data to the backend and database	1h	Rheaa
4	Manual testing and documentation	1h	Dane

Acceptance Criteria:

1. Given the user is interacting with the chatbot, when the user requests song recommendations, the chatbot should present prompts that clearly ask for their preferred Genre, Artist, Location, Album, and User emotion in a user-friendly manner.
2. Given the user is interacting with the chatbot, when the user requests song recommendations, there are suggested built-in default prompts that are available at the click of a button.
3. Given the user has provided their preferences through the prompts, when the chatbot sends the data to the backend and database, the chatbot should provide the user with a clear acknowledgment and confirmation that their preferences have been successfully submitted.
4. Given the user has provided their preference, when the user asks for recommendations, the chatbot should provide the user with a list of recommended songs.
5. Given the user has provided their preference, when the user clicks on a recommended song, the player should automatically play the song.

User Story #11c:

As a user, I would like to be able to automatically populate playlists per Genre, Artist, Location, Album and User emotion through PULSE chatbot

#	Description	Time	Owner
1	Build backend functions and routes to get required data from frontend (Research required)	3h	Rheaa
2	Build backend function(s) to automatically populate playlists per Genre	1.5h	Rheaa
3	Build backend function(s) to automatically populate playlists per Artist	0.25h	Rheaa
4	Build backend function(s) to automatically populate playlists according to popular music in a specific location (country or region)	0.25h	Rheaa
5	Build backend function(s) to automatically populate playlists according to user emotion	0.25h	Rheaa
6	Manual testing and testing documentation	1h	Noah

Acceptance Criteria:

1. Given the user's preference for a playlist based on Genre, when the user selects this option, the chatbot should present a user-friendly interface that allows them to choose a Genre and create a playlist with songs matching that Genre.
2. Given the user's preference for a playlist based on an Artist, when the user selects this option, the chatbot should provide a user-friendly interface that enables them to choose an Artist and create a playlist featuring songs by that Artist.
3. Given the user's preference for a playlist based on a Location, when the user selects this option, the chatbot should provide a user-friendly interface that enables them to choose a Location and create a playlist featuring songs by that Location.
4. Given the user's preference for a playlist based on an Album, when the user selects this option, the chatbot should provide a user-friendly interface that enables them to choose an Album and create a playlist featuring songs by that Album.
5. Given the user's preference for a playlist based on an User emotion, when the user selects this option, the chatbot should provide a user-friendly interface that enables them to choose an User emotion and create a playlist featuring songs by that User emotion.
6. Given the user's request for a playlist based on popular music in a specific location, when the user selects this option, the chatbot should provide an intuitive interface for specifying the region and generate a playlist featuring popular songs from that location.
7. Given the user's desire for a playlist based on their selected emotion, when the user selects this option, the chatbot should offer a user-friendly interface to select or describe their emotion, and then create a playlist tailored to that emotion.

User Story #11d:

As a user, I would like to be able to get song information by inputting a short phrase from the song into PULSE chatbot

#	Description	Time	Owner
1	Implementing a ChatGPT prompt and testing to make sure it has the required capabilities	2h	Rheaa
2	Implementing the functionality to let the user play the song once found	1h	Rheaa

3	(Backup) embedding functionality similar to chosic.com into the PULSE chatbot (Research Required)	4h	Rheaa
4	Manual testing and testing documentation	1h	Bodhi

Acceptance Criteria:

1. Given the user's desire to retrieve song information, when the user interacts with the chatbot, the chatbot should present a user-friendly prompt that allows the user to input a short phrase from the song.
2. Given the user has entered a short phrase from the song, when the user submits the phrase through the chatbot, then the chatbot should promptly process the input and provide relevant information about the song, such as its title, artist, and album.
3. Given the user's interaction with the chatbot, when the chatbot presents the song information, then the information should be presented in a clear and organized manner so that the user can easily read and understand it.
4. Given the user has imputed a short phrase, when the chatbot encounters difficulty in identifying the song based on the short phrase, then it should provide a clear and informative response to inform the user about the issue and, if possible, offer alternative steps or suggestions.
5. Given the user has implemented a short phrase, when the chatbot loads the song then, the user can play the song on spotify by clicking on it.

User Story #11e:

As a user, I would like to get help with finding/doing a certain action on the website by asking PULSE chatbot

#	Description	Time	Owner
1	Training the ChatGPT API to know specifics about our website features, structure, etc.	2.5h	Rheaa
2	Manually testing if the chatbot can provide all the needed assistance to the user	1h	Rheaa

Acceptance Criteria:

1. Given the user's need for assistance with website actions, when the user interacts with the chatbot, the chatbot should be able to understand and respond to a wide range of user queries related to website features, actions, and structure

2. Given the user's request for help, when the user asks the chatbot for assistance with a specific action or task on the website, the chatbot should provide clear and informative feedback.
3. Given the user's interaction with the chatbot, when the chatbot offers assistance, it should do so in a user-friendly and conversational manner, making the user feel comfortable and supported throughout the process.
4. Given the user has asked for which tab a feature should be in, when the chatbot responds, then the response should include the correct tab.
5. Given the user has asked to change a setting, when the chatbot responds, then the response should say whether the user can or cannot change said setting.

User Story #11f:

As a user, I would like to be able to give feedback to developers about requested features, bugs or general messages through PULSE chatbot

#	Description	Time	Owner
1	Training the Chatbot to identify when the user has something they want to communicate to the team, i.e, a feature, bug or general messages	1h	Rheaa
2	Creating functions to store these messages and have them sent to the project email (one of the team members personal email)	3h	Rheaa
3	Storing such messages/ interactions in the database	1h	Rheaa
4	Manual testing to ensure feedback is received by devs correctly	1h	Bodhi

Acceptance Criteria:

1. Given the user's desire to provide feedback, when the user interacts with the chatbot, the chatbot should be able to identify when the user wishes to communicate with the development team, whether it's regarding a feature request, a bug report, or a general message.
2. Given the user's intention to provide feedback, when the user expresses their feedback through the chatbot, the chatbot should facilitate a user-friendly and intuitive process for submitting feedback. This process should be clear and straightforward

3. Given the user's submission of feedback, when the user submits a message, the chatbot should provide a confirmation or acknowledgment that their feedback has been received and will be forwarded to the development team.
4. Given the user's expectation of responsiveness, when the chatbot stores the feedback, it should ensure that the messages are promptly forwarded to the appropriate project email address, which is one of the team member's personal email addresses.
5. Given the user has sent a feedback message, when the message is sent to the email, then this message should be the same as what is inputted to the chatbot.

Total Hours per Person:

Developer	Hours
Rheaa Sharma	36.45
Noah Stern	33
Dane Shaffer	30.5
Alex Polivka	31
Jason Zheng	30.75
Bodhi Scott	32.75
Total	**194.45**

Remaining Backlog:

Functional Requirements:

1. Basic Infrastructure
    a. As a user, I would like to be able to register for a PULSE account using my email (if time allows)
    b. As a user, I would like to be able to register for a PULSE account using OAuth such as Google (if time allows)
    ~~c. As a user, I would like to be able to register for a PULSE account using Spotify~~
    ~~d. As a user, I would like to be able to login and manage my PULSE account~~
    e. As a user, I would like to be able to reset my password if I forget it
    f. As a user, I would like to sync my account across devices/platforms
    ~~g. As a user, I would like to access all other pages from a home page~~
    h. Authentication and bad routing errors are handled gracefully and direct the user to an error page
2. Settings
    ~~a. As a user, I would like to have dark/light mode~~
    ~~b. As a user, I would like to be able to change text size~~
    ~~c. As a user, I would like to have custom backgrounds~~
    ~~d. As a user, I would like to have preset themes to change up the interface's look~~
    ~~e. As a user, I would like to be able to view my Pulse friends~~
    ~~f. As a user, I would like to edit my username~~
    ~~g. As a user, I would like to modify my profile information such as gender, profile picture and location.~~
3. Statistics Dump
    ~~a. As a user, I would like for my top songs and artists for the last 4 weeks, 6 months and all time to be tracked~~
    ~~b. As a user, I would like to be able to export Spotify data and import it into PULSE to get the following statistics~~
    ~~c. As a user, I would like for number of minutes listened to be tracked in my listening activity~~
    ~~d. As a user, I would like for percentage of total music listened to be tracked in my listening activity~~
    ~~e. As a user, I would like for times when music was listened to (morning, afternoon, night, etc.) in my listening activity~~

f. ~~As a user, I would like for times when songs were listened to fully or skipped or repeated in my listening activity~~
g. ~~As a user, I would like to to see songs, artists, playlists, genres and favorite era of music sorted by the metrics above~~
h. ~~As a user, I would like to track my follower numbers~~
i. ~~As a user, I would like for my most recently listened to songs to be displayed~~
j. ~~As a user, I would like for my songs, playlists, and albums that I've liked recently to be displayed~~
k. As a user, I would like to be able to specify what time period I want my stats to be from.
l. ~~As a user, I would like to be able to view statistics in multiple different graphs or charts~~
m. ~~As a user, I would like to be able to create custom layouts for the statistics page~~
n. As a user, I would like to be able to download my and my friend's statistics in a CSV and text file
o. As a user, I would like to be able to download my and my friend's statistics in a pdf file
p. ~~As a user, I would like to be able to create custom graphs with specified data and formats~~
q. ~~As a user, I would like to be able to have many choices of graphs to display data~~
r. ~~As a user, I would like to be able to view my friends' statistics as per previously mentioned user tracked statistics~~
s. As a user, I would like to import my friends stats from a csv/txt file to compare
t. ~~As a user, I would like to be able to view my playlists on the website~~
u. ~~As a user, I would like to see what genres I've listened to percentage-wise~~
v. ~~As a user, I would like to see what emotions I experience when listening to Spotify percentage-wise~~
w. ~~As a user, I would like to see what emotions I experience when listening to a selected song percentage-wise~~
x. As a user, I would like to be able to preview graphs before creating them
y. As a user, I would like to be able to modify graphs after they have been created

4. Playlist Page
    a. ~~As a user, I would like for recommendations for songs to add on the playlist based on Genre~~

~~b. As a user, I would like for recommendations for songs to add on the playlist based on Artist~~
c. As a user, I would like for recommendations for songs to add on the playlist based on Song Writer
d. As a user, I would like for recommendations for songs to add on the playlist based on Location
~~e. As a user, I would like for recommendations for songs to add on the playlist based on Album~~
~~f. As a user, I would like for recommendations for songs or playlists to based on User emotion presets (informed by Spotify API)~~
g. As a user, I would an AI to DJ for me based on BPM matching
~~h. As a user, I would like a playlist activity monitor that displays the amount of times the playlist was played in an user-friendly manner on the playlist page~~
i. As a user, I would like to choose two playlists and have a new playlist made from the Spotify-recommended songs of both playlists
j. As a user, I would like to choose two playlists and have a new playlist to merge them and have them flow into each other based on BPM
~~k. As a user, I would like to be able to set my emotion parameters using a user chosen playlist.~~
~~l. As a user, I would like to manage playlists through Pulse~~
~~m. As a user, I would like to auto generated playlists through Pulse based on genre~~

5. Song Page
   ~~a. As a user, I would like to be able to find similar songs (based on Spotify API parameters such as loudness) to the current song~~
   ~~b. As a user, I would like to be able to find similar songs by specifying Spotify API target parameters such as loudness to a chosen song.~~
   c. As a user, I would like to be able to view lyrics to songs
   d. As a user, I would like for songs to be automatically captioned based on lyrics (If time allows)
   e. As a user, I would like for songs to be automatically removed from playlists after a certain threshold of skips set by the user
   f. As a user, I would like for the song to have music videos automatically linked if available on Youtube or other such platforms
   ~~g. As a user, I would like for a song activity monitor that displays the amount of times the song was played in an user-friendly manner on the song page~~
6. Game Page

   a. ~~As a user, I would like to play the game "Guess the Song/Artist" as a quiz format and heads-up format~~
   b. ~~As a user, I would like to play games such as Guess who listens to this song as a quiz format and heads-up format~~
   c. ~~As a user, I would like to play the game "Guess the Next Lyric" as a quiz format and heads-up format~~

7. Search (if time allows)
   a. ~~As a user, I would like to be able to search for songs, albums, artists, playlists.~~
   b. As a user, I would like to be able to filter my search with keywords such as year, genre, new songs and popularity.
8. Song Player (if time allows)
   a. ~~As a user, I would like to be able to listen to samples of songs when they are recommended to me~~
   b. ~~As a user, I would like to be able to control Spotify playback through a music player on PULSE~~
   c. As a user, I would like to start Spotify playback through a Spotify device on PULSE after searching for a playlist or song
   d. As a user, I would like to manage the Spotify queue by searching a song and adding or removing it
9. Music Uploader (if time allows)
   a. As a user, I would like to be able to upload local files and add them to new/existing playlists
   b. As a user, I would like to be able to auto-fill metadata for these local files
10. TicketMaster API (if time allows)
   a. As a user, I would like to be able to view music show recommendations when viewing relevant songs/artists/albums
11. PULSE Chat Bot (if time allows)
   a. ~~As a user, I would like to interact with a PULSE chatbot through a textbox~~
   b. ~~As a user, I would like to be able to get song recommendations per Genre, Artist, Location. Album and User emotion through PULSE chatbot~~
   c. ~~As a user, I would like to be able to automatically populate playlists per Genre, Artist, Location. Album and User emotion through PULSE chatbot~~
   d. ~~As a user, I would like to be able to get song information by inputting a short phrase from the song into PULSE chatbot~~
   e. ~~As a user, I would like to get help with finding/doing a certain action on the website by asking PULSE chatbot~~
   f. ~~As a user, I would like to be able to give feedback to developers about requested features, bugs or general messages through PULSE chatbot~~

Non-Functional Requirements:

I. Quality of Life
   A. As a user, I would like the website to be responsive with reasonable speed and at all times
      1. Reasonable speed: < 400ms for typical navigation inputs
      2. Be able to reasonably handle double-inputs

II. Accessibility
   A. As a user, I would like the website to be usable in multiple formats
      1. Support for landscape and portrait mode
      2. Support for desktop and mobile browser
      3. Support for scaled-up/down windows

III. Maintainability
   A. As a developer, I would like to be able to easily maintain features
   B. As a developer, I would like to be able to easily add new features

IV. Security
   A. Personal information and passwords will be encrypted when transferred using an 18-bit encryption
   B. Encrypt using a more-secure Caesar cipher (if time allows)
   C. User information from Spotify should be stored in a secure manner to protect against common exploits such as cross-site-scripting, buffer overflow, and SQL injection using industry standard practices such as Microsoft's own SQL Server documentation on common security practices. This includes practices like input sanitization, privilege controls, and encryption

V. Architecture and Performance
   A. As a developer, I want to run the frontend using ReactJS and the backend using Python
   B. As a developer, I want to use SQL for storing data
   C. As a developer, I want to connect the frontend and backend using Flask