Milestones List:

1) Finalize the framework and APIs that will be used in the development of the project
    a) Decide as a group what framework we will continue development with. This framework will also need to support the APIs we will be using for our project to continue successfully.
2) Decide if a database is needed, and if so, implement a backend for it.
    a) Explore the possibilities of our project and figure out if a database is needed. If it is, then time will be needed to implement a backend to support it.
3) Perform setup of a of proxy server to enable API testing
    a) A proxy server will need to be set up in order to enable us to test the API effectively.
4) Set up the skeleton of the application in the desired framework
    a) Set up the skeleton of the application so that we can begin testing API calls and testing in a sandbox type environment.
5) Begin implementing the user interface and linking it with the rest of the application
    a) Create the user interface and start testing its functionality with the rest of the application.
6) Create the first playlist using the application
    a) Develop the first functional iteration of the application capable of generating a playlist.
7) Create a playlist that meets the standards/requirements of our group using the application
    a) Develop and refine the program until it is able to create a playlist that contains no songs we want to skip
8) Complete testing program for the application
    a) Develop and execute the test plan for the application, including testing creating playlists under various conditions and with people outside the group

Timeline:

Task	**Start Date**	**Completion Date**
Research and investigate Spotify API calls	October 2024	October 2024
Research AI usage	October 2024	October 2024

Research frameworks to use for development of the application	October 2024	October 2024
Research mobile app development	October 2024	October 2024
Identify all APIs that will be used to collect data for playlists and get required credentials for all	October 2024	October 2024
Create app in Spotify to get API credentials	November 2024	November 2024
Create proxy server so API calls to Spotify can be made	November 2024	December 2024
Develop user interface that users can interact with to get their playlists	November 2024	March 2024
Develop AI model that helps decide what songs are included in a user's playlist	November 2024	January 2025
Develop a test plan for the application overall	March 2025	March 2025
Develop logic to create personalized playlists	January 2025	March 2025
Develop code to centralize data collection for use when developing playlists	November 2024	March 2025
Develop documentation for the application as the process moves forward	November 2024	March 2025

Develop a unit testing scheme	March 2025	April 2025
Execute testing at the end of development to test all functionality	March 2025	April 2025

Effort Matrix:

Task	**Effort Breakdown**
Research and investigate Spotify API calls	Austin: 30% Derrick: 10% Joe: 60%
Research AI usage	Austin: 80% Derrick: 10% Joe: 10%
Research frameworks to use for development of the application	Austin: 10% Derrick: 80% Joe: 10%
Research mobile app development	Austin: 10% Derrick: 10% Joe: 80%
Identify all APIs that will be used to collect data for playlists and get required credentials for all	Austin: 60% Derrick: 20% Joe: 20%
Create app in Spotify to get API credentials	Austin: 50% Derrick: 25% Joe: 25%
Create proxy server so API calls to Spotify can be made	Austin: 50%

	Derrick: 25% Joe: 25%
Develop user interface that users can interact with to get their playlists	Austin: 25% Derrick: 25% Joe: 50%
Develop AI model that helps decide what songs are included in a user's playlist	Austin: 25% Derrick: 50% Joe: 25%
Develop a test plan for the application overall	Austin: 10% Derrick: 80% Joe: 10%
Develop logic to create personalized playlists	Austin: 30% Derrick: 40% Joe: 30%
Develop code to centralize data collection for use when developing playlists	Austin: 60% Derrick: 20% Joe: 20%
Develop documentation for the application as the process moves forward	Austin: 25% Derrick: 25% Joe: 50%
Develop a unit testing scheme	Austin: 10% Derrick: 10% Joe: 80%
Execute testing at the end of development to test all functionality	Austin: 30% Derrick: 40% Joe: 30%