# Technical Design Document

## 1. Feature Name:

Filtering Out Playlist Tracks from Search Results.

## 2. Feature Description:

Modify the search functionality to filter out tracks that are already present in the current playlist from appearing in the search results.

## 3. Prerequisites:

- Knowledge of React.  
- Understanding of the existing components and their interactions in the project.

## 4. Requirements:

- When a user searches for tracks, the application should not display the tracks that are already present in the user's playlist.  
- The user interface remains responsive during the filtering operation.

## 5. Feature Components & Responsibilities:

- Spotify.js Component: Modify the search method to filter out the tracks that are already present in the current playlist.  
- SearchResults.js Component: Ensure that the component can handle and display the filtered list of tracks correctly.

## 6. Implementation Details:

1. Modify the Search Method in Spotify.js:  
 - After fetching the search results, compare each track with the tracks in the current playlist.  
 - Remove any tracks from the search results that are already present in the playlist before returning the results.  
  
2. Handle the Filtered List in SearchResults.js:  
 - No major modifications may be needed if this component is already set up to handle and render a list of tracks. Just ensure that it can correctly handle and display an updated list of tracks.

## 7. Reasoning:

- Filtering out the already added tracks from the search results will enhance the user experience by preventing users from adding duplicate tracks to the playlist.  
- Implementing this feature requires minor modifications to existing components, making it a relatively straightforward enhancement with a significant impact on usability.

## 8. Testing:

- After implementation, conduct thorough testing to ensure that already added tracks do not appear in the search results.  
- Verify that the application's existing functionality remains intact and that the user interface is responsive during the operation.

## 9. Risks:

- Incorrect implementation could lead to filtering out the wrong tracks or not filtering out the tracks at all.  
- Performance issues might arise when handling extensive playlists and search results.