Feature Proposal: Share Playlists

November 6th 2024

OBJECTIVE

The objective of this feature is to enable users to share the playlists they create through SpotiLists in a variety of ways, including sharing as a link or as an image file.

BACKGROUND

Currently, SpotiLists allows users to search for songs and save them to a playlist on Spotify. However, it lacks a sharing feature that would let users easily share their curated playlists with others. Adding sharing options will enhance the user experience, potentially increasing the app's appeal and encouraging more users to engage with it.

This feature will introduce "Share" buttons within the playlist view, allowing users to share their playlists in different formats. Sharing as a link will generate a unique URL that takes the recipient to a dedicated page displaying the playlist in a visually engaging way, where they can also save it to their own Spotify account. Sharing as an image file will let users download an aesthetically formatted image of their playlist, which they could then post on social media platforms like Instagram.

TECHNICAL DESIGN

Share Button Components

Instead of a single share button with a menu, the sharing functionality will be split into two distinct buttons displayed at the bottom of the Playlist component in Playlist.js. The first button, labeled "Share as Link," will handle generating and displaying a unique link to the playlist. The second button, labeled "Share as Image," will trigger image generation of the playlist, formatted for sharing.

Link Sharing

Clicking the “Share as Link” button will create a new, shareable page on the site, using the playlist data stored at the time of sharing. Here’s how the link-sharing process will work:

1. Generate Link and Save Data:

- When “Share as Link” is clicked, a new page URL will be generated with a unique playlist ID to represent this specific playlist.
- The playlist’s name, tracklist, and relevant data will be saved in a server-side database or lightweight backend storage, so the data persists and can be accessed from the generated URL. This will allow a new viewer to access the playlist data without requiring a Spotify API call.

2. New Shared Playlist Page:

- The generated link will point to a new page, PlaylistViewPage, which will retrieve the saved playlist data using the unique playlist ID in the URL.
- This new page will display the playlist name and tracklist in a layout similar to the main playlist view, with a “Save to Spotify” button at the top for users to save the playlist to their account.
- This “Save to Spotify” button can reuse the existing save functionality from Spotify.js for consistency.

This approach ensures that playlist data is stored at the time of sharing, allowing the playlist to be viewed later without re-fetching data from the Spotify API.

Image Sharing

The “Share as Image” button will use html2canvas to capture an image of only the playlist’s tracklist and name, making it suitable for social media sharing. Here’s how this process will work:

1. Image Generation:

- When “Share as Image” is clicked, html2canvas will capture only the portion of the Playlist component that includes the tracklist and playlist name. Additional CSS or a specific wrapper around the tracklist section can be used to ensure that only the intended content is captured.

2. Download Image:

- Once captured, the image will be prepared for download as a .png file. The filename will default to something meaningful, such as Playlist_<PlaylistName>.png, to make organization easy for users.

- A “Download” button will appear to allow users to save the image file to their device.

Additional State and Methods in App.js

The App.js component will manage the new functionality through a few additional state variables and methods. Key variables include a boolean, isImageCaptureReady, to control the image generation flow, and sharedPlaylistUrl, which holds the generated link to the playlist view page.

The following methods will handle the main actions:

- handleLinkShare(): Generates the unique link by saving the playlist data and updates sharedPlaylistUrl. This link will be copied to the clipboard.

- handleImageShare(): Triggers html2canvas to capture the tracklist and playlist name section, then presents the image for download.

CAVEATS

Image Capture for Large Playlists

Capturing large playlists as an image may result in layout and readability issues. For example, a long playlist may be difficult to fit within a single image, resulting in either small, unreadable text or the need for multiple images. To mitigate this, the app could limit the number of tracks captured or provide a “scrollable” image capture option that breaks up the playlist across multiple images. However, implementing multi-image capture could complicate the sharing process and increase file size, which may not be ideal for all users.

Storage and Performance Considerations for Link Sharing

Storing playlist data on the server for link sharing introduces considerations around storage limits, particularly if users frequently create and share new playlists. Over time, these playlist records could take up considerable storage. Implementing a cleanup mechanism to remove unused or expired links could help maintain storage efficiency, but this would add backend complexity.

Alternatively, a serverless storage solution (e.g., Firebase, AWS S3) could be explored, although such solutions could also introduce additional costs as the number of saved playlists grows.

Consistency and Compatibility Across Devices

Features like automatic copying to clipboard and html2canvas may behave differently on various devices and browsers. For example, clipboard access may be limited on older or mobile browsers, and image capture may not be pixel-perfect on all screen sizes. To address this, extensive cross-device testing will be necessary, especially on mobile devices where behavior may vary. An alternative would be to provide users with a direct “Copy Link” option if clipboard access fails, though this would slightly impact user convenience.