\*\*Rework the database so that every artist/song has to be searched from Spotify. Then, upon object creation, store the Spotify object ID to facilitate lookup when someone logs in with Spotify to search for live performance of songs/artists that they like.\*\*

-Doing this now for songs. Verify it works and then replicate efforts for artists.

-Add in verification that artistSpotifyId and selectedSpotifyObject.artists[0].id are the same before allowing creation of a new object.

fix clear state actions so they don't return initialState, just clear the specific portion of the state object

make it so <Link> tags are yellow when hovered over (search results).

**\* Add ability to login to your Spotify account \*:**

* **Get a list of top artist/songs and pull relevant performances**
  + **https://developer.spotify.com/console/get-current-user-saved-tracks/**
* **Add a button next to live performances that adds songs to your Spotify playlist**
* **Spotify dashboard: https://developer.spotify.com/dashboard**

1. Fix the login functionality - fixed login not working, now need to make it so reload doesn't destroy local storage token

2. Add clear state actions for every view that utilizes state

3. Remove the votes field from the User schema and create a new schema associated with a user and a performance to enhance lookup speed

4. Build out artist info page

5. Change voting so that performance objects store vote totals, but the actual votes themselves are associated with users. This way, when searching if a user has votes on a performance, you can do a dictionary lookup rather than the current method of looping through every single vote object. OR just change the performance voteList to be an object with user ID as the key and vote as the value.

6. Redesign TopPerformanceTable – split into two columns, make thumbnail images smaller, on right column, add submitted by information along with user-provided description of performance.

7. Allow for venues to be searched – e.g. see all performances at Red Rocks. Add in descriptions and history of venues.