|  |
| --- |
| Faculty of Applied Sciences and Technology |
| ITC5202 - Project |
|  |
| **Gurpreet Kaur Lakshmi Priyanka Annapureddy**  **Submission Date** |
| **12/11/23** |

|  |
| --- |
| This document explains how to process api calls …………………………. |

Table of Contents

[Step – 1: Index/Home Page 2](#_Toc153179930)

[Step 2 – Get Albums 4](#_Toc153179931)

[Step 3 – Get Artists 6](#_Toc153179932)

[Step 4 – Get Shows 8](#_Toc153179933)

[Step 5 – Get Recommendations 10](#_Toc153179934)

[Step 6 – Get Profile 12](#_Toc153179935)

[Step 7 – Search For an Artist/Album 14](#_Toc153179936)

[Step 8 – Create Playlist 17](#_Toc153179937)

[End Points 17](#_Toc153179938)

[Summary 18](#_Toc153179939)

# Step – 1: Index/Home Page

This is the starting page of the website. Once user run the project, the following page will be appeared (index.html)

A screenshot of a music album

Description automatically generated

The JavaScript code you provided performs the following major steps for get new releases.

**Event Listener for DOMContentLoaded:**

* Use document.addEventListener to wait for the DOM to be fully loaded before executing the script.

**Get Token Function:**

* Define an async function (getToken) to retrieve the Spotify access token using the client ID and client secret.
* Make a POST request to Spotify's token endpoint and parse the resulting JSON to get the access token.

**Fetch New Releases:**

* Call getToken to obtain the access token.
* Use the token to make a request to the Spotify API endpoint for new releases.
* Parse the response and create HTML cards for each album, displaying relevant information.
* Append the cards to the albumsContainer.

**Create Card Function:**

* Define a function (createCard) to create an HTML card for each album with details such as image, album name, release date, total tracks, artist, and type.

**Get Profile Button Click Event:**

* Add an event listener to the button with the ID getProfileBtn to initiate the Spotify authorization process.

**Authorization URL Generation:**

* Construct the authorization URL for Spotify using client ID, response type, redirect URI, and scopes.
* Redirect the user to the Spotify authorization page.

**Extract Access Token from URL:**

* Extract the access token from the URL hash after the user is redirected back from the Spotify authorization page.

**Fetch User Details:**

* If an access token is available, use it to make an API request to fetch the user details from Spotify.

**Create Profile Card:**

* Define a function (createProfileCard) to generate an HTML card displaying user profile details.
* Update the albumsContainer with the user's profile card.

**Authorize Button Click Event:**

* Add an event listener to the button with the ID authorizeBtn to handle Spotify authorization for creating playlists

# Step 2 – Get Albums

The JavaScript code you provided performs the following major steps for get albums

**Authorization and Token Retrieval:**

* When the page loads, the event listener is set for the "Authorize Spotify" button (authorizeBtn).
* On button click, it redirects the user to the Spotify authorization URL, where the user can log in and authorize your application.
* After authorization, it extracts the authorization code from the URL and exchanges it for an access token using the Spotify API.

**Featured Albums Retrieval:**

* It fetches a list of featured albums from Spotify using the obtained access token.
* For each album in the response, it creates an HTML card containing album details and appends it to the albums-container.

**Playlist Creation:**

* If an authorization code is present in the URL (indicating the user has authorized the app), it exchanges it for an access token.
* Using the obtained access token, it creates a new public playlist named "Test" with a description "test-xml."
* It then fetches details of the created playlist and updates the HTML to display the playlist details.

**Adding Songs to the Playlist:**

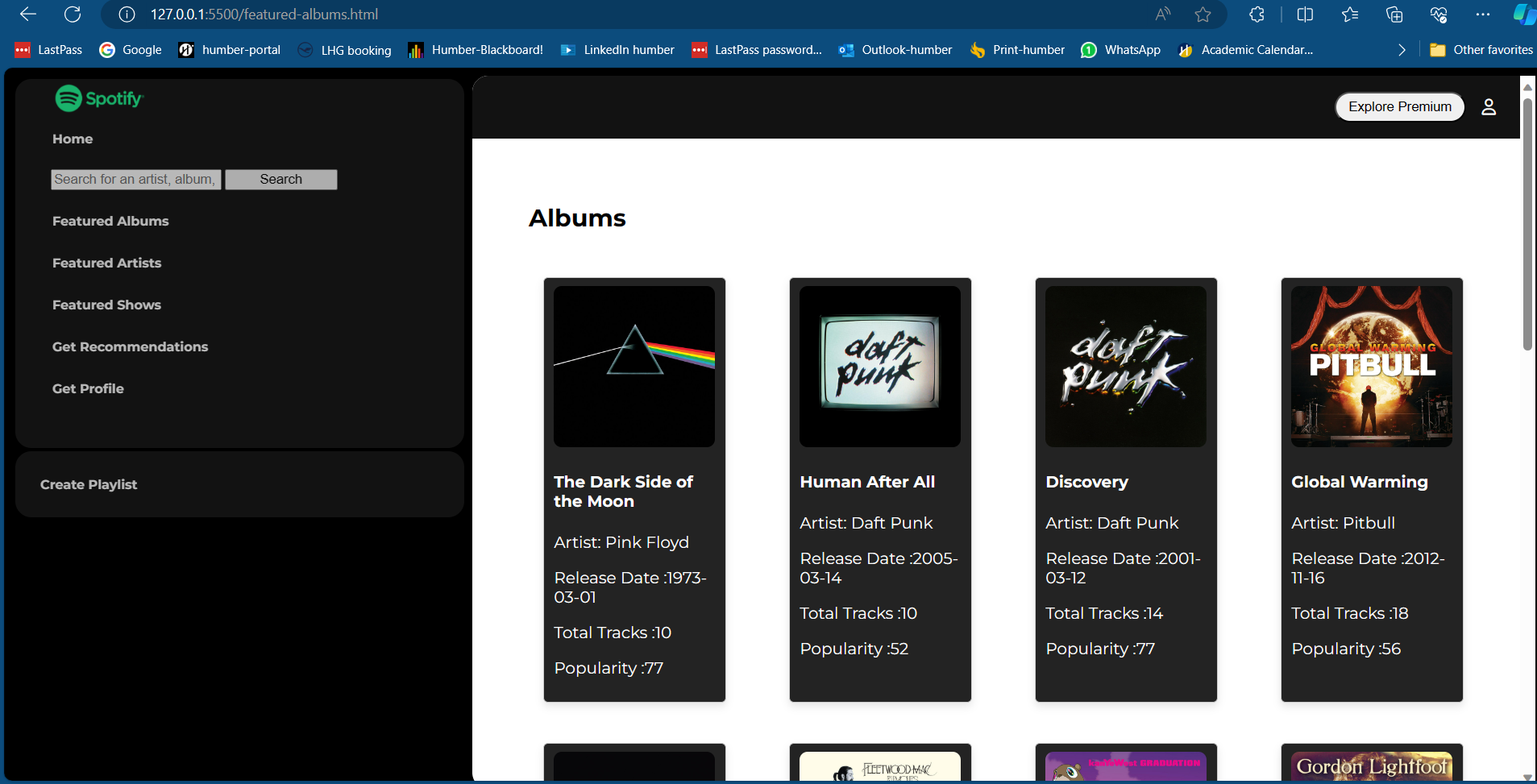
* It adds three specific songs to the created playlist using the obtained access token.

**Fetching and Displaying Playlist Details:**

* It fetches details of the created playlist again after a short delay (using the sleep function).
* It updates the HTML to display various details of the playlist, such as name, link, total tracks, owner's display name, owner's type, public status, and the date the first track was added.

To test this program, open the HTML file in a web browser and click on Featured Albums in the left side bar navigation menu.

**Output screenshot –**

****

# Step 3 – Get Artists

The JavaScript code you provided performs the following major steps for get artists

**Fetching Artist Information:**

* Obtain the access token using the getToken function.

Specify artist IDs and construct an API URL to fetch information about multiple artists.

**API Request - Fetching Artist Data:**

* Use the fetch function to make an API request to Spotify's endpoint for artists. Include the access token in the headers.

**Creating Artist Cards:**

* For each artist in the retrieved data, create an artist card using the createArtistCard function and append it to the DOM.

**Creating Artist Card Function:**

* Define a function (createArtistCard) to take an artist object, create a card, and populate it with artist details.

**Displaying Artist Details:**

* Populate the artist card with an image, name, popularity, and follower count.

**Create Playlist Button and Authorization:**

* Add an event listener to a button with the ID authorizeBtn for creating playlists. This button triggers Spotify authorization.

**Playlist Creation Logic:**

* If there's an authorization code in the URL, exchange the code for an access token and call the createPlaylist function.

**Playlist Creation Function:**

* Define a function (createPlaylist) to make a POST request to create a playlist and log the result.

To test this program, open the HTML file in a web browser and click on Featured Artists in the left side bar navigation menu.

**Output screenshot –**

A screenshot of a computer

Description automatically generated

# Step 4 – Get Shows

The JavaScript code you provided performs the following major steps for get shows

**Fetching Featured Albums:**

* Obtain the access token using the getToken function.
* Construct the API URL to fetch information about featured albums from Spotify.

**API Request - Fetching Album Data:**

* Use the fetch function to make an API request to Spotify's endpoint for shows (which represent albums in this case). Include the access token in the headers.

**Creating Album Cards:**

* For each album in the retrieved data, create an album card using the createCard function and append it to the DOM.

**Creating Album Card Function:**

* Define a function (createCard) to take an album object, create a card, and populate it with album details.

**Displaying Album Details:**

* Populate the album card with an image, title, artist name, total episodes, show type, and publisher.

**Create Playlist Button and Authorization:**

* Add an event listener to a button with the ID authorizeBtn for creating playlists. This button triggers Spotify authorization.

**Playlist Creation:**

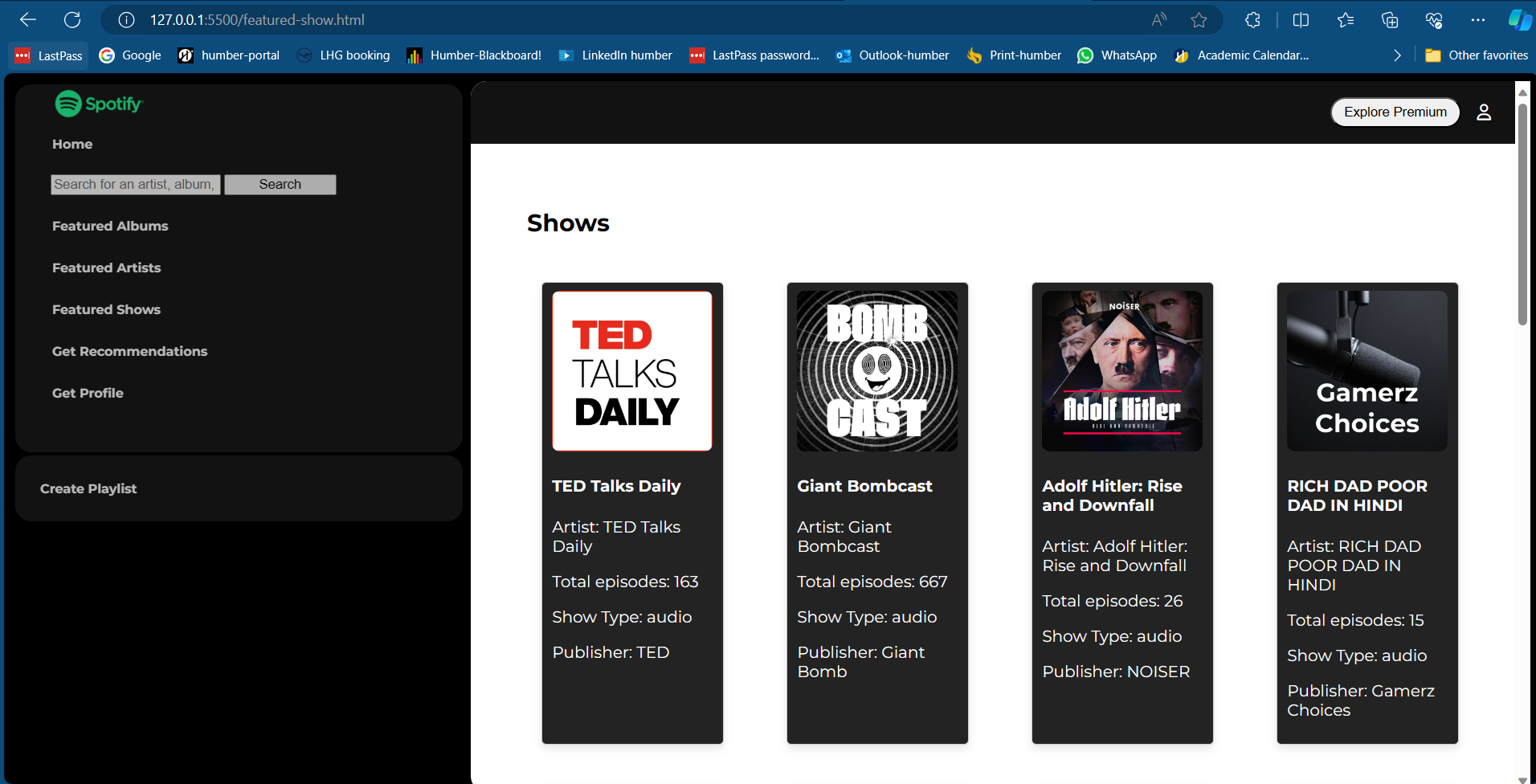
* If there's an authorization code in the URL, exchange the code for an access token and call the createPlaylist function.

**Playlist Creation Function:**

* Define a function (createPlaylist) to make a POST request to create a playlist and log the result.

To test this program, open the HTML file in a web browser and click on Featured Shows in the left side bar navigation menu.

**Output Screenshot –**



# Step 5 – Get Recommendations

The JavaScript code you provided performs the following major steps for get recommendations.

**Token Retrieval Function:**

* Create an asynchronous function (getToken) to retrieve the access token needed for API requests using the client credentials flow.

**Fetching Featured Albums:**

* Obtain the access token using the getToken function.
* Construct the API URL to fetch recommendations from Spotify based on seed artists, genres, and tracks.

**API Request - Fetching Album Data:**

* Use the fetch function to make an API request to Spotify's recommendations endpoint. Include the access token in the headers.

**Creating Album Cards:**

* For each album in the retrieved data, create an album card using the createCard function and append it to the DOM.

**Creating Album Card Function:**

* Define a function (createCard) to take an album object, create a card, and populate it with album details.

**Displaying Album Details:**

* Populate the album card with an image, title, artist name, and any other relevant details.
* Create Playlist Button and Authorization:
* Add an event listener to a button with the ID authorizeBtn for creating playlists. This button triggers Spotify authorization.

**Playlist Creation Logic:**

* If there's an authorization code in the URL, exchange the code for an access token and call the createPlaylist function.

**Playlist Creation Function:**

* Define a function (createPlaylist) to make a POST request to create a playlist and log the result.

To test this program, open the HTML file in a web browser and click on Get Recommendations in the left side bar navigation menu.

**Output Screenshot –**

A screenshot of a computer

Description automatically generated

# Step 6 – Get Profile

The JavaScript code you provided performs the following major steps for get profile

**Authorization URL Generation:**

* Construct the authorization URL for Spotify using client ID, response type, redirect URI, and scopes.
* Redirect the user to the Spotify authorization page.

**Extract Access Token from URL:**

* Extract the access token from the URL hash after the user is redirected back from the Spotify authorization page.

**Fetch User Details:**

* If an access token is available, use it to make an API request to fetch the user details from Spotify.

**Create Profile Card:**

* Define a function (createProfileCard) to generate an HTML card displaying user profile details.

**Display User Profile:**

* Append the created profile card to the albumsContainer to display user details.

**Add New Playlists Button Click Event:**

* Add an event listener to the button with the ID authorizeBtn to initiate the process of creating new playlists.

**Exchange Authorization Code for Access Token:**

* If there's an authorization code in the URL, exchange it for an access token by making a POST request to Spotify's token endpoint.

**Create Playlist:**

* Define a function (createPlaylist) to make a POST request to create a new playlist on Spotify.

**Fetch Playlist Details:**

* After creating a playlist, fetch its details using the access token.

**Add Songs to Playlist:**

* Define a function (addSongsToPlaylist) to add songs to the created playlist.

**Sleep Function:**

* Define a sleep function to introduce a delay before fetching playlist details to ensure the playlist is fully updated.

**Fetch Playlist Details Again:**

* Fetch the playlist details again after a short delay to display the updated information.

**Display Playlist Details:**

* Define a function (displaygetplayDetails) to create HTML elements and display playlist details like name, description, total tracks, owner details, etc.
* Append these elements to the playlistContainer to display the information

To test this program, open the HTML file in a web browser and click on Get Profile in the left side bar navigation menu.

**Output Screenshot –**

A screenshot of a computer

Description automatically generated

# Step 7 – Search For an Artist/Album

The JavaScript code you provided performs the following major steps for search for an artist/album.

**Event Listener for DOMContentLoaded:**

* Use document.addEventListener to wait for the DOM to be fully loaded before executing the script.

**Search Functionality:**

* Get references to the search input, search button, and results container.
* Add an event listener to the search button to initiate a Spotify search when clicked.
* Retrieve the search term from the input, trim it, and call the searchSpotify function if it's not empty.

**Search Spotify Function:**

* Define the searchSpotify function that takes a query and fetches results from the Spotify API.
* Call the getToken function to obtain the access token.
* Construct the Spotify API URL with the search query and make a GET request with the access token.
* Display the search results by calling the displayResults function.

**Get Token Function:**

* Define an asynchronous getToken function that retrieves the Spotify access token using client credentials.
* Make a POST request to Spotify's token endpoint with the client ID and client secret.
* Parse the response to obtain the access token.

**Display Results Function:**

* Define the displayResults function that takes the Spotify API response and displays the search results.
* Check the response structure and log it for debugging.
* Clear the resultsContainer and create HTML cards for each album.
* Append the album cards to the resultsContainer.

**Create Album Card Function:**

* Define the createAlbumCard function to generate an HTML card for each album in the search results.
* Extract album details such as image, name, artist, release date, and total tracks.
* Create HTML elements for each detail and append them to the album card.

**Authorize Button Click Event:**

* Add an event listener to the button with the ID authorizeBtn to handle Spotify authorization for creating playlists.

**Exchange Authorization Code for Access Token:**

* Extract the authorization code from the URL if present.
* If an authorization code is available, exchange it for an access token by making a POST request to Spotify's token endpoint.

**Create Playlist Function:**

* Define the createPlaylist function to make a POST request to create a new public playlist named 'Test'.
* Log the details of the created playlist and fetch its details using the fetchPlaylistDetails function.

**Fetch Playlist Details Function:**

* Define the fetchPlaylistDetails function to fetch details of a playlist using the access token and playlist ID.
* Call the addSongsToPlaylist function to add predefined songs to the created playlist.
* Call the fetchgetplayDetails function to fetch the playlist details again after a short delay.

**Add Songs to Playlist Function:**

* Define the addSongsToPlaylist function to add predefined songs to the created playlist using the access token and playlist ID.

**Sleep Function:**

* Define a sleep function to introduce a delay before fetching playlist details to ensure the playlist is fully updated.

**Fetch Playlist Details Again Function:**

* Define the fetchgetplayDetails function to fetch playlist details after a short delay and call displaygetplayDetails to update the HTML.

**Display Playlist Details Function:**

* Define the displaygetplayDetails function to create HTML elements and display playlist details like name, description, total tracks, owner details, etc.
* Append these elements to the playlistContainer to display the information.

To test this program, open the HTML file in a web browser and click on search in the left side bar navigation menu.

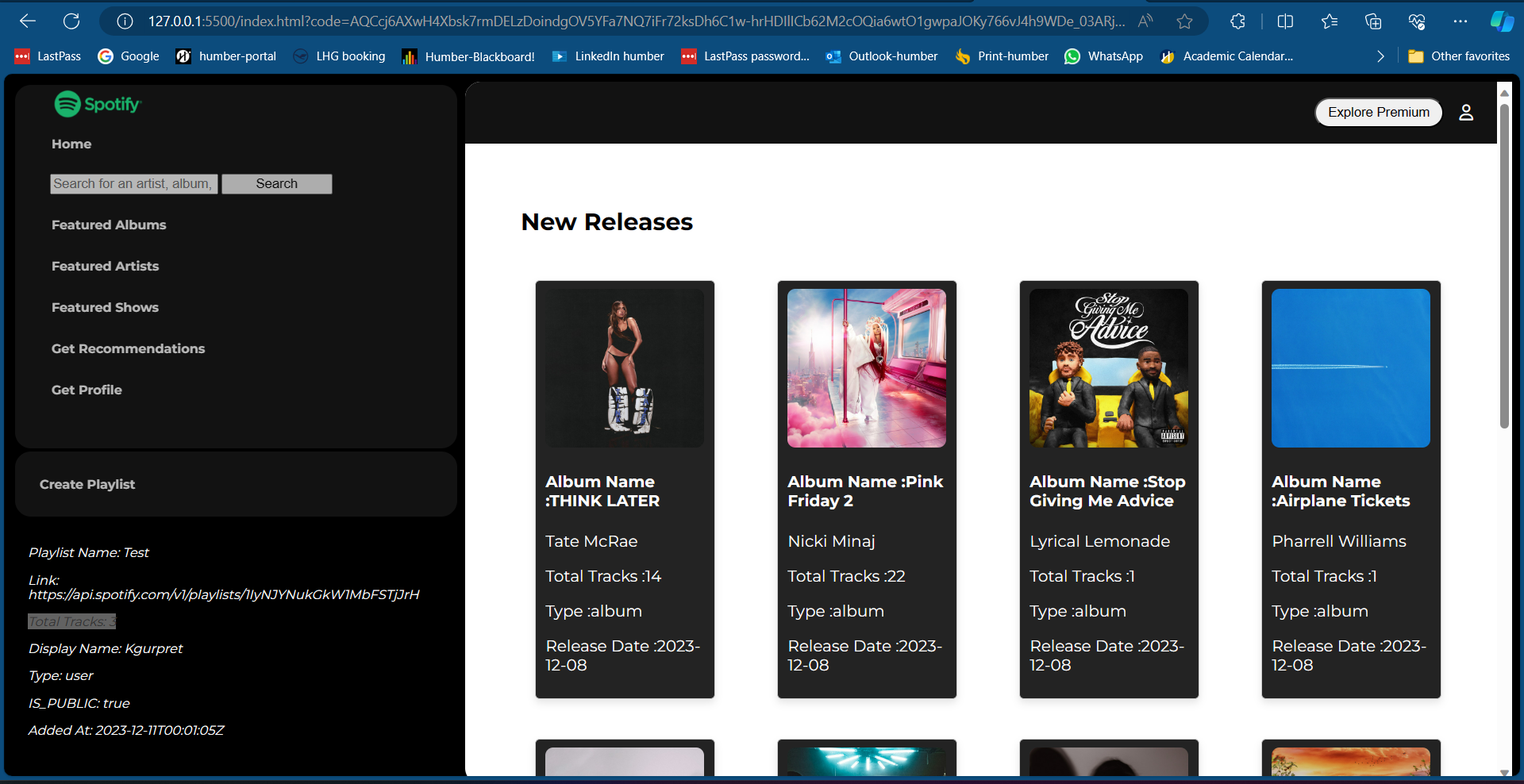
**Output Screenshot –**

Searched for Arjit Singh

A screenshot of a music website

Description automatically generated

# Step 8 – Create Playlist



# End Points

1. **Get New Releases:**

* Endpoint: <https://api.spotify.com/v1/browse/new-releases>
* Description: Retrieve a list of new album releases on Spotify.

1. **Get Albums:**

* Endpoint: <https://api.spotify.com/v1/albums>
* Description: Fetch details about specific albums on Spotify. The API may support fetching multiple albums based on their IDs.

1. **Get Artists:**

* Endpoint: <https://api.spotify.com/v1/artists>
* Description: Retrieve information about specific artists on Spotify. This may involve fetching details about multiple artists based on their IDs.

1. **Get Shows:**

* Endpoint: <https://api.spotify.com/v1/shows>
* Description: Fetch information about specific shows on Spotify. In this context, shows might represent albums.

1. **Get Recommendations:**

* Endpoint: <https://api.spotify.com/v1/recommendations>
* Description: Obtain music recommendations from Spotify based on seed artists, genres, and tracks.

1. Get Profile:

* Endpoint: <https://api.spotify.com/v1/me>
* Description: Retrieve details about the authenticated user's Spotify profile.

1. **Search for an Artist/Album:**

* Endpoint: <https://api.spotify.com/v1/search>
* Description: Search Spotify for artists, albums, tracks, or playlists based on a given query.

1. **Create Playlist:**

* Endpoint: <https://api.spotify.com/v1/me/playlists>
* Description: Create a new playlist for the authenticated user on Spotify.

1. **Add Songs to Playlist:**

* Endpoint: [https://api.spotify.com/v1/playlists/{playlist\_id}/tracks](https://api.spotify.com/v1/playlists/%7bplaylist_id%7d/tracks)
* Description: Add specific songs to an existing playlist on Spotify.

1. **Playlist Details:**

* Endpoint: [https://api.spotify.com/v1/playlists/{playlist\_id}](https://api.spotify.com/v1/playlists/%7bplaylist_id%7d)
* Description: Retrieve detailed information about a specific playlist on Spotify.

# Summary

As a team, we initially outlined a plan to address ten specific endpoints within the project. We divided the APIs evenly between two team members, delving into the corresponding documentation and conducting thorough testing using Postman.

Throughout the implementation process, we encountered several challenges:

- Some APIs necessitated a premium membership for full functionality.

- Retrieving details for specific APIs, such as creating and fetching playlists, involved consecutive calls and required incorporating a designated sleep time to ensure accurate results.