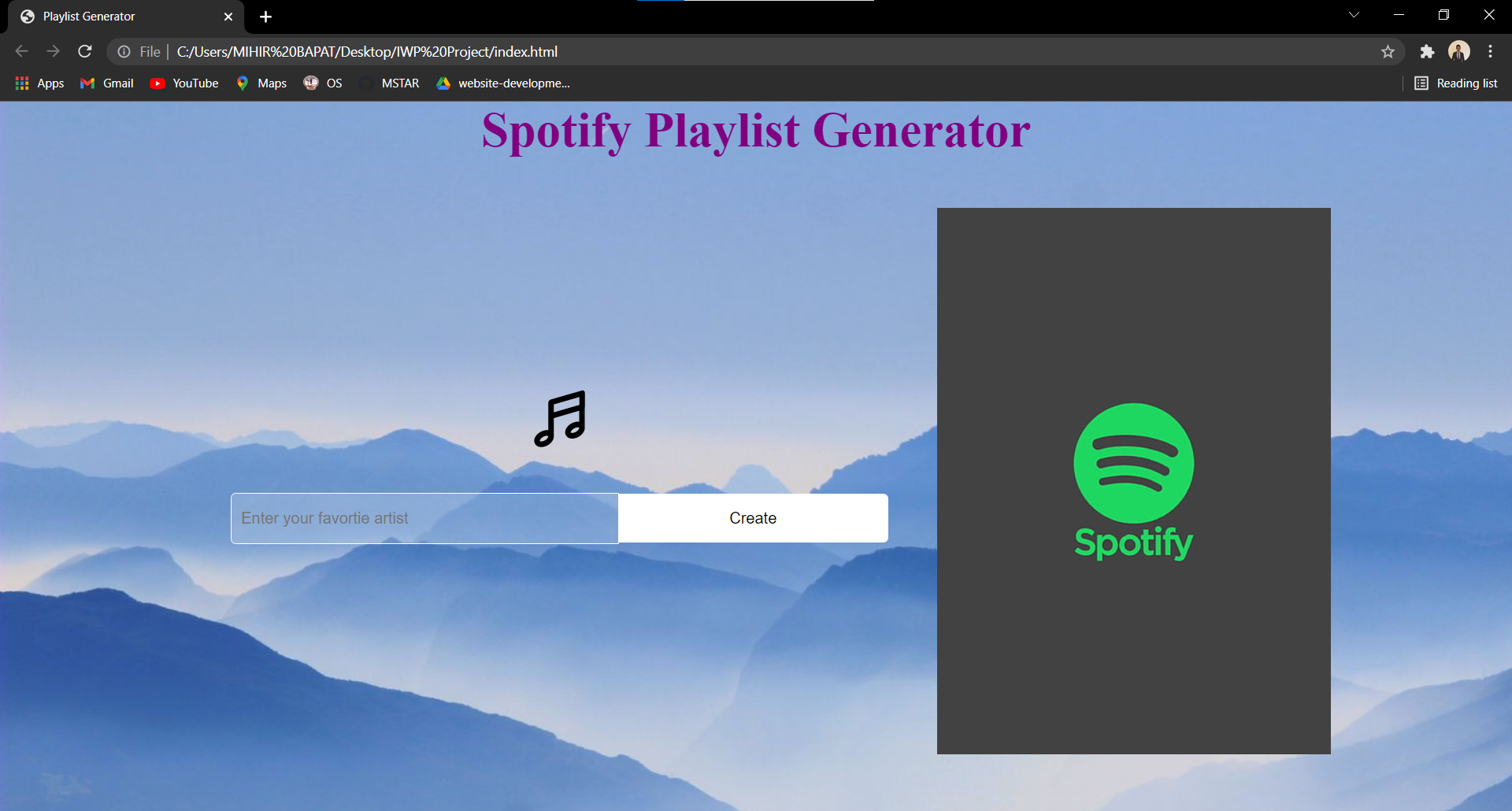
**IWP Project Report: Spotify Playlist Generator**

**Introduction**

Spotify is one of the most popular music streaming applications in the world. This project deals with interacting with the Spotify web APIs in order to generator an automated playlist consisting of songs from our favorite Artists.

**Preview**

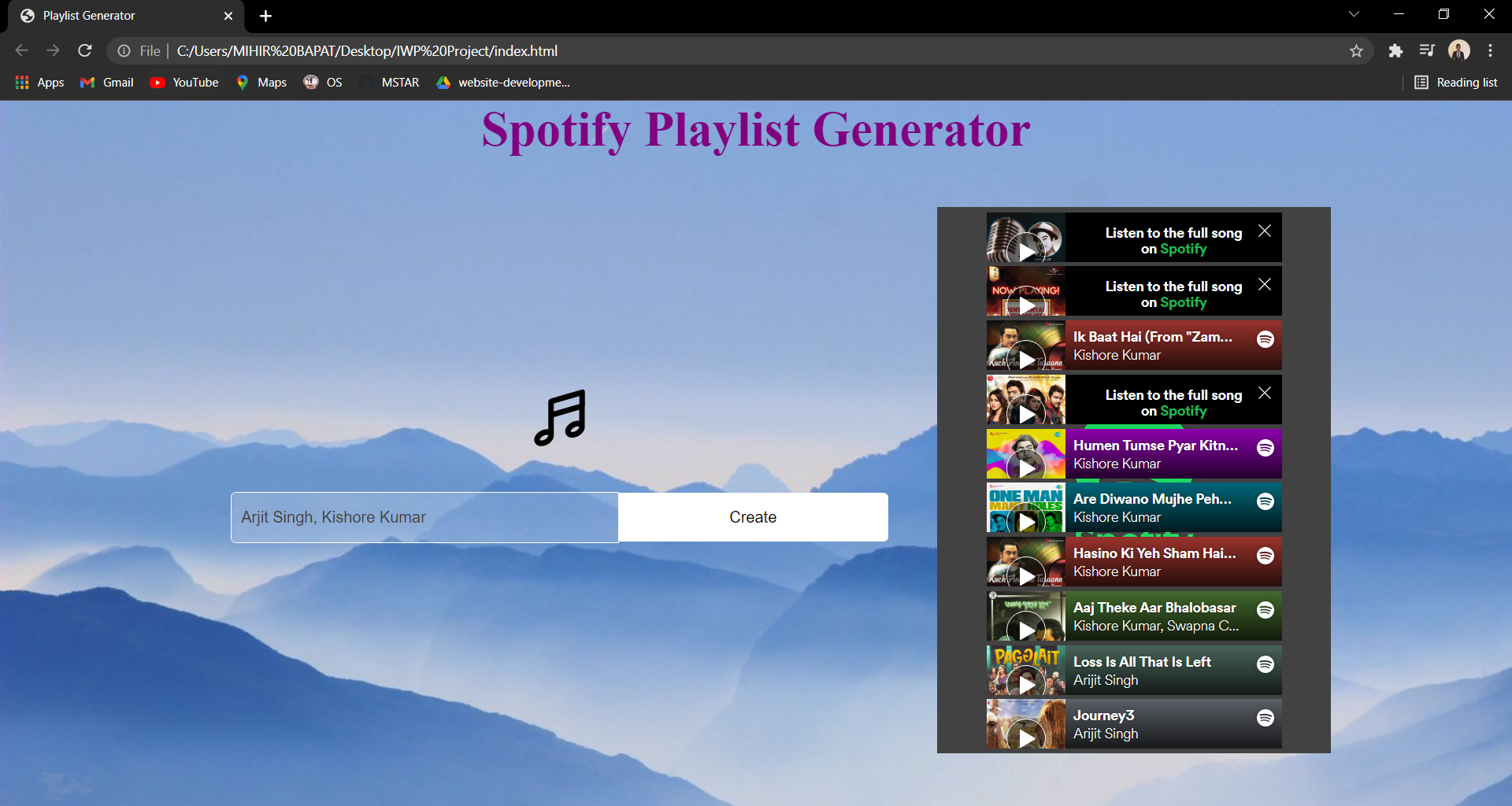
****

Application Default Webpage

The webpage consists of a basic layout consisting of an input bar and Spotify playlist bar which is empty by default. The input bars consist of a placeholder text which says the users to enter the names of their favorite Artists. Users are allowed to enter multiple names into this box and the different Artist names have to be separated by a comma (,).

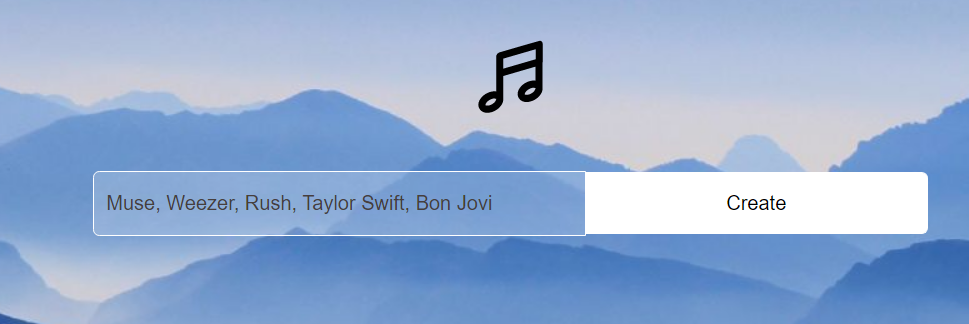
Following the addition of Artist names, the user needs to click the create button and a playlist of 10 songs from their favorite artists is created in the Spotify playlist bar. Users can then play these songs by clicking the play button or may choose to hit the Spotify icon next to any given song and open/play the song on the Spotify app instead.

**Test Case**



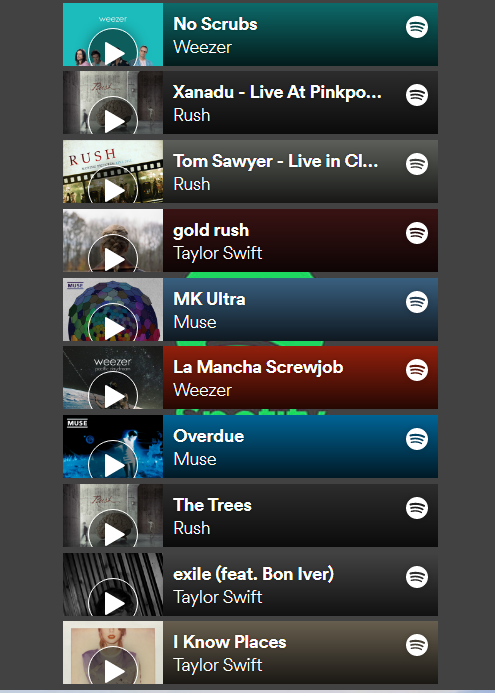
Application Webpage when a Query is passed

In the above example, multiple artists (2) are entered into the input box separated by a comma (,). After clicking the create button, the application runs the JavaScript to interact with the Spotify API and proceeds to create a playlist of 10 tracks from your favorite artists.



Input Box with Multiple Artists

Users are allowed to enter multiple artists as many as they wish to in the input box as long as they are separated by a comma (,). The application will pick all the artists’ names and select some of their songs to generate the playlist.



Playlist consisting of songs from multiple artists

**Technologies Used and Code**

The project is purely based on HTML, CSS and JavaScript. The HTML and CSS files are fairly simple and are used to generate the outline and template of the application webpage. The application functionalities are defined via the JavaScript file.

Inside the JavaScript file, there are four functions making actual calls to the Spotify API while multiple other support and helper functions to ensure the smooth flow and working of the script.

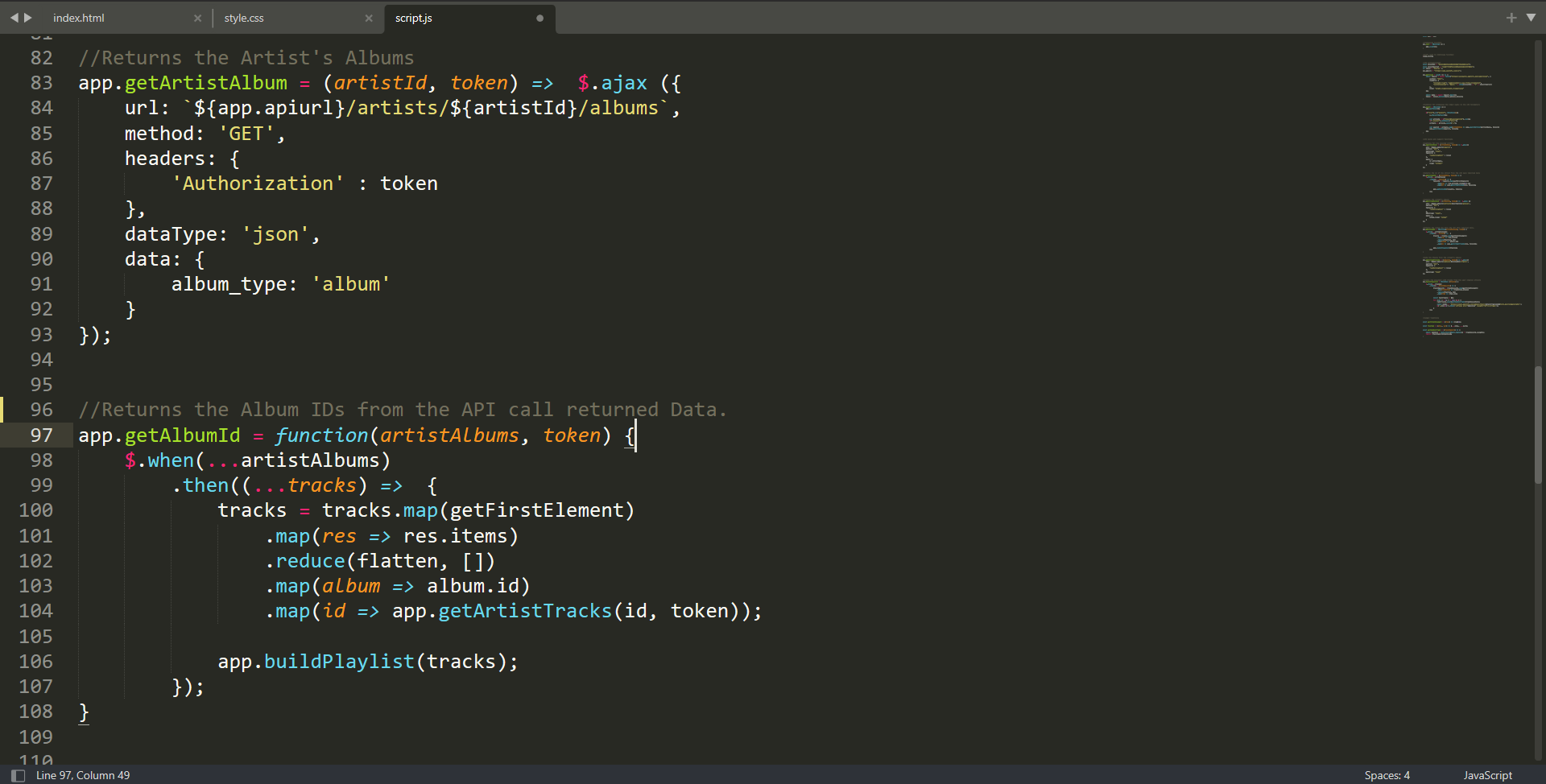
API Calls: The following functions interact with the Spotify API for various API calls.

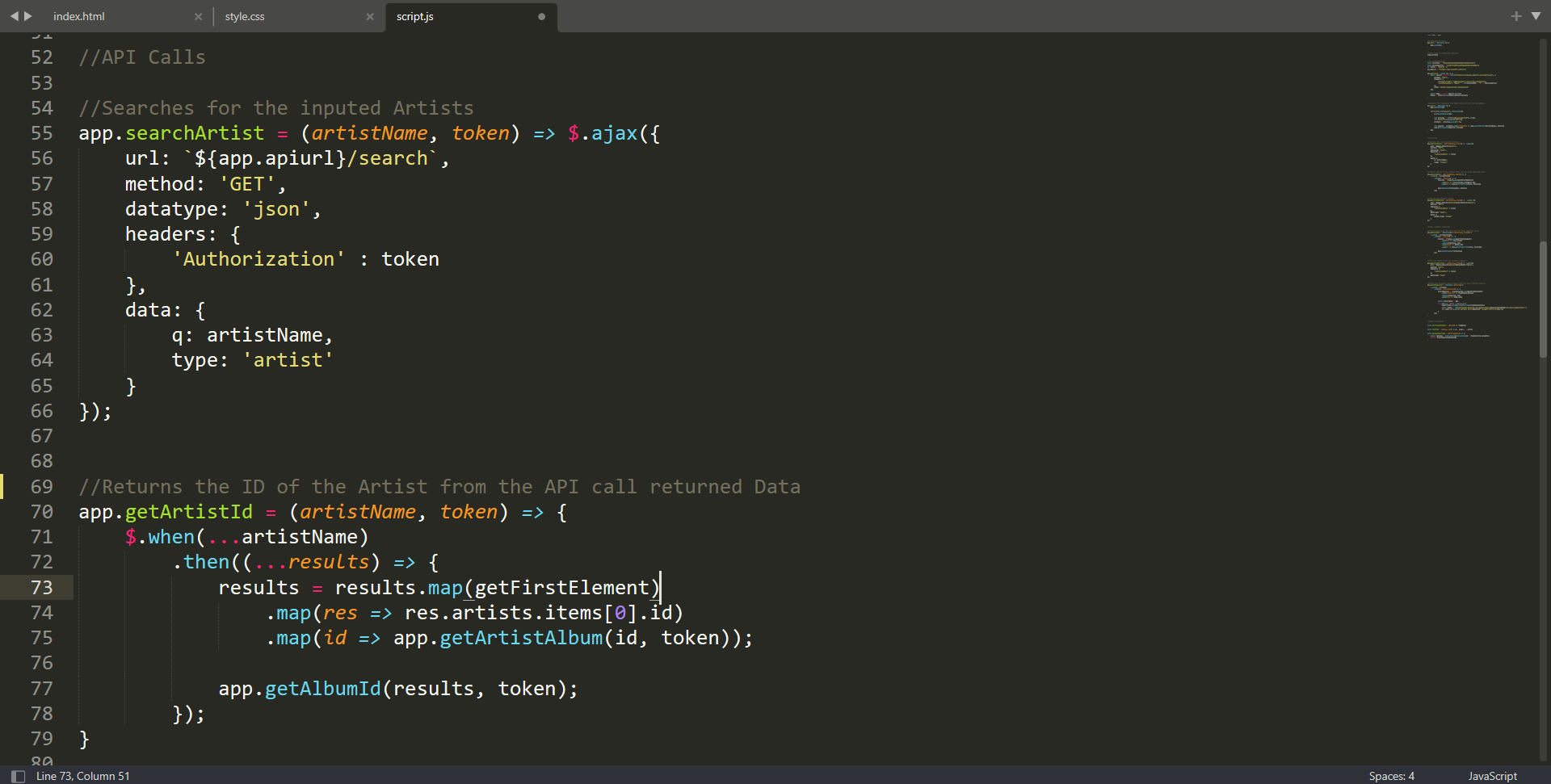
app.getToken(): is used to generate the API token required for making further API calls. API Method: POST

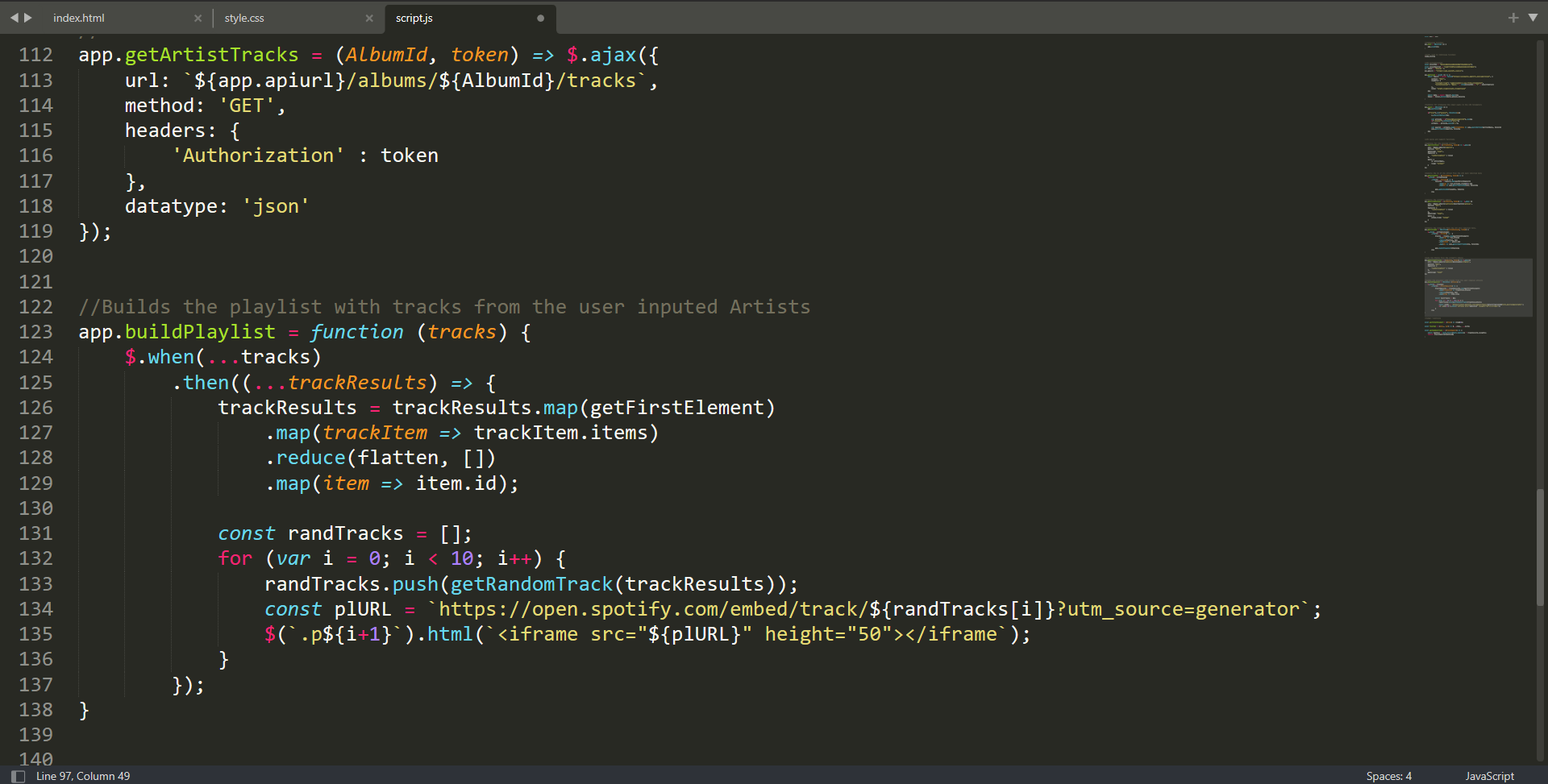
app.searchArtist(): is used to search for the artist based on the input passed query. API Method: GET

app.getArtistAlbum(): is used to get the Albums of our selected Artists using their IDs. API Method: GET

app.getArtistTracks(): is used to get the tracks from an album of our selected Artist using the album IDs. API Method: GET

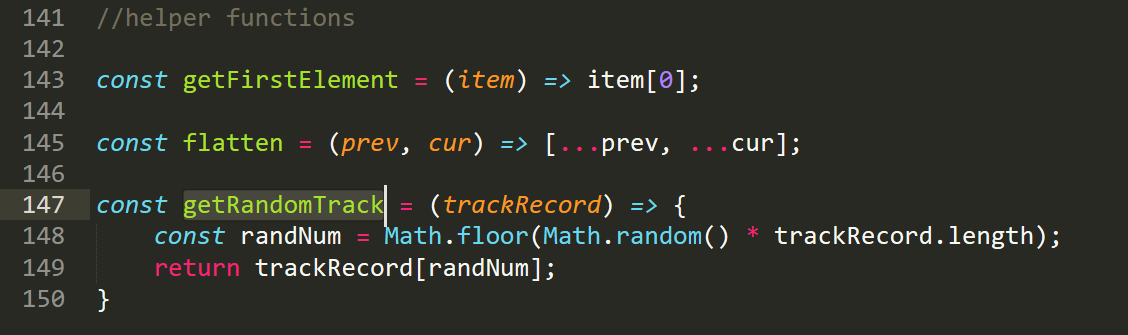
JS Functions Code Snippets 





The additional function used in between the API calls maintain the proper flow of the Application Working.

Finally, for creating the playlist app.buildPlaylist() function is used which stacks the all the tracks from various albums from all the selected Artists and uses the getRandomTrack() helper function to select random tracks of our favorite artists to generate the playlist. The app.buildPlaylist() function then passes these selected tracks to a Spotify URL to create their embedded widgets on our webpage in order to play these songs.



Helper Functions

**Enclosures**

The submission zip file will consist of the following things:

1. Report.docx (current file): The project report and details on its implementations.
2. README.dat: Readme file with instructions on how to run and view the project and any prerequisites or installations required (if any).
3. Demo.mp4: A video file with a short demo of the working of the web application.
4. Explanation Video.mp4: A video explaining the entire working of the project and how it was performed and coded.
5. Contributions.txt: A text file consisting of all my contributions or tasks performed during the working of this project.
6. Spotify Playlist Generator: The folder consisting of all the files regarding the actual application. These include:

Index.html: The HTML file consisting of the template. The application can be opened by running this file through a browser.

Style.css: A CSS file with all the styling elements for the neat look of the template.

Script.js: A JavaScript that interacts with Spotify API and performs all the functionalities of this application.

Images: A folder consisting of the images used in the template.