

searchSong Function

- **Purpose:** This function searches for songs based on user input.
1. `let songName = document.getElementById('search-field').value;`
 - Retrieves the value from an input field with the ID `search-field` and stores it in the variable `songName`.
 2. `fetch('https://api.lyrics.ovh/suggest/${songName}')``
 - Sends a GET request to the Lyrics.ovh API to get song suggestions based on the input `songName`.
 3. `.then(res => res.json())`
 - Converts the response to JSON format.
 4. `.then(data => displaySong(data.data));`
 - Passes the array of song data to the `displaySong` function.

fetchSongs Function

- **Purpose:** This function fetches a list of top songs.
1. `fetch('https://api.lyrics.ovh/suggest/top')`
 - Sends a GET request to fetch top song suggestions from the Lyrics.ovh API.
 2. `.then(res => res.json())`
 - Converts the response to JSON format.
 3. `.then(data => displaySong(data.data));`
 - Passes the song data to the `displaySong` function.

fetchArtistSongs Function

- **Purpose:** This function fetches songs by a specific artist.
1. `fetch('https://api.lyrics.ovh/suggest/${artistName}')``
 - Sends a GET request to get song suggestions for a specific artist.
 2. `.then(res => res.json())`
 - Converts the response to JSON format.
 3. `.then(data => displaySong(data.data.slice(0, 30)));`
 - Passes up to 30 songs to the `displaySong` function.

resetPage Function

- **Purpose:** Clears the song display area.
1. `const songContainer = document.getElementById("song-container");`
 - Selects the element with ID `song-container`.
 2. `songContainer.innerHTML = '';`
 - Clears the inner HTML, removing all displayed songs.

`displaySong` Function

- **Purpose:** Displays a list of songs on the webpage.
1. `const songContainer = document.getElementById("song-container");`
 - Selects the element where songs will be displayed.
 2. `songContainer.innerHTML = '';`
 - Clears any existing song data to prepare for new content.
 3. `songs.forEach(song => { ... });`
 - Iterates over each song in the `songs` array.
 4. `const songDiv = document.createElement("div");`
 - Creates a new `div` element for each song.
 5. `songDiv.className = "div1";`
 - Assigns a CSS class name to the `div` for styling.
 6. `songDiv.innerHTML = ...`
 - Sets the HTML content of the `div` to display song details, including album cover, title, artist, and an audio preview.
 7. `songContainer.appendChild(songDiv);`
 - Adds the `div` to the `songContainer` on the webpage.

This code effectively allows users to search for songs, view top songs, or browse songs by a specific artist using the Lyrics.ovh API, displaying the results dynamically on a webpage.