

# Assignment 5: Artist Search

EECS 493 Winter 2024

Released: **3/17/2024**

Due: **4/7/2024** at 11:59pm

## Submission Instructions

- Please submit your work to Canvas as a zip file, named "a5\_<your\_uniqname>.zip".
  - Remove the angle bracket when naming your zip.
  - Renaming (e.g., "-1") done by Canvas is fine.
- This zip file should contain a *single directory* containing *all files and directories* provided in the starter code.
  - i.e. The zip file should have the following structure:

```
a5_uniqname.zip
  └── a5_uniqname
      ├── README.txt (OPTIONAL)
      ├── img
      │   ├── 1.jpg
      │   └── 2.jpg
      ├── index.css
      ├── index.html
      ├── loading.gif
      └── script.js
```
  - Please keep the existing directory structure and do not add any more folders/files (other than the README.txt for bonus credit).
- *Not following the submission instructions will result in a penalty.*

## Assignment Objective

The objective of this assignment is for you to gain practical experience with the MVC pattern using Vue.js, a web application development framework, and Bootstrap, a CSS framework for responsive design.

## Project Overview Video

[Assignment 5 Intro Video](#)

*Remark: The search results from the video may not be up to date. Please refer to the screenshots below for more recent, accurate results.*

## **Grading Breakdown**

We would like you to create a digital “Artist Search” website as outlined in the attached spec below. The application has 6 main components (denominator of 100 points):

1. Search Bar - *25 points*
2. Artists Grid - *30 points*
3. Navigation Tabs - *5 points*
4. Track Information - *5 points*
5. Genre Selection - *20 points*
6. Sort Menu - *15 points*
7. **[Optional]** Bonus Features - *up to 10 points*

## **Starter code**

You will use the starter code we provide to finish this assignment: [a5\\_w24\\_starter\\_code.zip](#)

The starter code we provide includes the following files:

1. script.js: Placeholder for *one* Vue.js instance [Put your JS code in this file]
2. index.html: HTML page [Put your HTML code in this file]
3. loading.gif: Show this gif while waiting for artist information to come back [Optional]
4. index.css: Additional customized style [Do NOT modify this file]
5. img/1.jpg & img/2.jpg: Two image placeholders [Your final code won't need them]

## **Remarks:**

- You aren't required to use any of the starter code, but it's there to help you.
- Unless you know what you are doing, please do NOT modify anything inside the <head> tag in index.html.
- Please refer to Piazza for any modifications and clarifications.
- Make sure that your application (webpage) behaves properly on the latest version of Google Chrome. Your graders will use Chrome.
- To make sure your code runs correctly, it might be helpful to test it on a different computer.

## Screenshots

Homepage with search bar:

493 Artist Search

Enter the name of the artist and enter :

Search Result (Total 0 Found.)

List of results

Sort list of results by ▾

After searching for an artist:

493 Artist Search

Enter the name of the artist and enter :

Search Result (Total 50 Found.)

ALL Electronic Dance Pop

List of results

Sort list of results by ▾

Description Track Info

**Artist Name:** ayokay & Quinn XCII  
**Collection Name:** 4ft to Infinity - EP  
**Price:** 4.99  
**Kind:** song  
**Preview:** [Link](#)

Description Track Info

**Artist Name:** ayokay & Quinn XCII  
**Collection Name:** Kings of Summer  
- Single  
**Price:** 1.29  
**Kind:** song  
**Preview:** [Link](#)

Description Track Info

**Artist Name:** ayokay  
**Collection Name:** In the Shape of a Dream  
**Price:** 8.99  
**Kind:** song  
**Preview:** [Link](#)

Description Track Info

**Artist Name:** ayokay  
**Collection Name:** The Shine (feat. Chelsea Cutler) - Single  
**Price:** 1.29  
**Kind:** song  
**Preview:** [Link](#)

## "Track Info." tab for the left two results:

493 Artist Search

Enter the name of the artist and enter :

Search Result (Total 50 Found.)

**ALL** Electronic Dance Pop

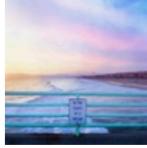
List of results

Sort list of results by: ▾

	Description	Track Info
	Track ID: 1160446542	
	Country: USA	

	Description	Track Info
	Artist Name: ayokay & Quinn XCII	
	Collection Name: Kings of Summer	
	- Single	
	Price: 1.29	
	Kind: song	
	Preview: <a href="#">Link</a>	

	Description	Track Info
	Track ID: 1433987966	
	Country: USA	

	Description	Track Info
	Artist Name: ayokay	
	Collection Name: The Shine (feat. Chelsea Cutler) - Single	
	Price: 1.29	
	Kind: song	
	Preview: <a href="#">Link</a>	

## Sort menu:

493 Artist Search

Enter the name of the artist and enter :

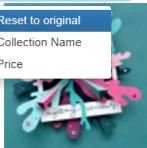
Search Result (Total 50 Found.)

**ALL** Electronic Dance Pop

List of results

Sort list of results by: ▾

- Reset to original
- Collection Name
- Price

	Description	Track Info
	Track ID: 1160446542	
	Country: USA	

	Description	Track Info
	Artist Name: ayokay & Quinn XCII	
	Collection Name: Kings of Summer	
	- Single	
	Price: 1.29	
	Kind: song	
	Preview: <a href="#">Link</a>	

	Description	Track Info
	Track ID: 1433987966	
	Country: USA	

	Description	Track Info
	Artist Name: ayokay	
	Collection Name: The Shine (feat. Chelsea Cutler) - Single	
	Price: 1.29	
	Kind: song	
	Preview: <a href="#">Link</a>	

## After sorting by “Collection Name”:

493 Artist Search

Enter the name of the artist and enter :

Search Result (Total 50 Found.)

**ALL** Electronic Dance Pop

List of results

Sort list of results by: ▾

	<a href="#">Description</a> <a href="#">Track Info.</a> <b>Artist Name:</b> ayokay & Quinn XCII <b>Collection Name:</b> 4ft to Infinity - EP Price: 4.99 Kind: song <a href="#">Preview: Link</a>		<a href="#">Description</a> <a href="#">Track Info.</a> <b>Artist Name:</b> ayokay <b>Collection Name:</b> 4ft to Infinity - EP Price: 4.99 Kind: song <a href="#">Preview: Link</a>
	<a href="#">Description</a> <a href="#">Track Info.</a> <b>Artist Name:</b> ayokay <b>Collection Name:</b> 4ft to Infinity - EP Price: 4.99 Kind: song <a href="#">Preview: Link</a>		<a href="#">Description</a> <a href="#">Track Info.</a> <b>Artist Name:</b> ayokay <b>Collection Name:</b> 4ft to Infinity - EP Price: 4.99 Kind: song <a href="#">Preview: Link</a>

## After sorting by “Price”:

493 Artist Search

Enter the name of the artist and enter :

Search Result (Total 50 Found.)

**ALL** Electronic Dance Pop

List of results

Sort list of results by: ▾

	<a href="#">Description</a> <a href="#">Track Info.</a> <b>Artist Name:</b> ayokay <b>Collection Name:</b> Clocks - Single Price: 0.99 Kind: song <a href="#">Preview: Link</a>		<a href="#">Description</a> <a href="#">Track Info.</a> <b>Artist Name:</b> Disco Lines & ayokay <b>Collection Name:</b> I Got You - Single Price: 0.99 Kind: song <a href="#">Preview: Link</a>
	<a href="#">Description</a> <a href="#">Track Info.</a> <b>Artist Name:</b> ayokay <b>Collection Name:</b> On My Mind - Single Price: 0.99 Kind: song <a href="#">Preview: Link</a>		<a href="#">Description</a> <a href="#">Track Info.</a> <b>Artist Name:</b> ayokay <b>Collection Name:</b> All Mine - Single Price: 1.29 Kind: song <a href="#">Preview: Link</a>

After (resetting sort to original and) selecting the “Dance” and “Pop” genre selections:

493 Artist Search

Enter the name of the artist and enter :

Search Result (Total 22 Found.)

[ALL](#) [Electronic](#) [Dance](#) [Pop](#)

List of results

Sort list of results by: ▾

	<a href="#">Description</a>	<a href="#">Track Info</a>		<a href="#">Description</a>	<a href="#">Track Info</a>
<b>Artist Name:</b> ayokay & Nightly <b>Collection Name:</b> Digital Dreamscape <b>Price:</b> 9.99 <b>Kind:</b> song <b>Preview:</b> <a href="#">Link</a>			<b>Artist Name:</b> ayokay <b>Collection Name:</b> All Mine - Single <b>Price:</b> 1.29 <b>Kind:</b> song <b>Preview:</b> <a href="#">Link</a>		
	<a href="#">Description</a>	<a href="#">Track Info</a>		<a href="#">Description</a>	<a href="#">Track Info</a>
<b>Artist Name:</b> ayokay <b>Collection Name:</b> On My Mind - Single <b>Price:</b> 0.99 <b>Kind:</b> song <b>Preview:</b> <a href="#">Link</a>			<b>Artist Name:</b> ayokay <b>Collection Name:</b> I Still Need You - Single <b>Price:</b> 1.29 <b>Kind:</b> song <b>Preview:</b> <a href="#">Link</a>		

## Ground Rules

1. The primary point of the assignment is to use MVC pattern in developing the application using Vue.js. Therefore, reference and direct modification of HTML elements (such as `$(".class")`, `append()`, `document`) in JavaScript and jQuery are **NOT allowed**.
  - a. For example, using jQuery or JavaScript code that access HTML elements is **NOT allowed** (will result in significant deductions).
2. We've given you some starter code, and you'll only need to write JS code inside Vue instance(s) (in the JS file) and Vue directives in the HTML file.
  - a. We already created one Vue instance for you. You may add other instances if you prefer, but **NO** extra JS code should be written outside of Vue instances.
3. Every style-related element of the assignment **MUST** be done via Bootstrap. You are **NOT allowed** to use inline styling or define/use any custom CSS class/id other than the ones we provided in index.css.
  - a. i.e. Do **NOT** modify index.css-
4. The provided index.html file includes (links to) jQuery, Bootstrap, Vue.js, and Axios.
  - a. In order for index.html to work properly, make sure you
    - i. Have internet connection,
    - ii. Do **NOT modify** anything inside the `<head>` tag in index.html.
  - b. Note that index.html has some hard-coded data. This is just to show you an example, but the hard-coded data should eventually be replaced.
5. Follow the submission instructions at the beginning of the spec.
6. After unzipping your submission, the index.html file should work by directly opening it in Google Chrome, as long as there is an internet connection.
7. This assignment uses the iTunes API, which again requires internet access. Plan ahead if you will be in a situation that has no internet access. It is your responsibility to read the documentation and figure out the data structure of the JSON object.
8. This assignment heavily requires online search and reading the documentation of Vue.js and iTunes API that we use. The assignment may include materials that are not covered in class. We believe that, in today's programming practice, searching and mining information from online materials is a significant part of a programmer's expertise.
9. Check out Piazza frequently for any changes or additional information.

## Requirements

We outline requirements for each of the game components below. **Everything listed below, unless labeled as “Suggested”, is required.** Please be sure to examine the provided video closely to understand the expected behavior. You are allowed to change anything in “index.html” and “script.js”.

**Please see the Hints & Resources section for really helpful hints & resources!**

### General

- a. Ground rules - *max 50 points off if not followed*
1. **Search Bar (30 points)**
  - a. There is one input textbox provided in the HTML file.
  - b. When a user types a keyword and presses the “Enter” key, the entered string should be used to make an AJAX call that requests a json object.
    - i. Hint: Use v-on:keyup to check if the key was “enter (keycode:13)” or not
  - c. The entered string will be used to search artists whose names contain the keyword.
  - d. Do NOT use jQuery \$.ajax, but instead use Vue.js AJAX call – Axios – or simply use fetch().
  - e. The URL that you need to make an AJAX request and the data structure of the returned json object are specified here:
    - i. iTunes Search API: <https://goo.gl/UXwDce>
    - ii. When sending the request, all media types & entities should be included.
    - iii. When a query has over 50 results, it should only return a maximum of 50 tracks.
  - f. **Print out the returned json object in the console, using JS console.log(), when making requests for the APIs.**
    - i. This should be the *only* thing that is printed in the console.
    - ii. We will take off points if the above requirements are not met.
  - g. When there are no artists returned from iTunes, alert the user, using JS alert(), that no artist was found with the keyword.
  - h. “Total *N* found” message should display the total number of artists (*N*) returned by the iTunes API.
2. **Artists Grid (30 points)**
  - a. The returned artists should be displayed in *two columns*, where each result grid contains both an image and 5 types of related information displayed right next to each other (when the browser is in full size).
  - b. One exception will be the last row that may have less than two artist grids depending on the number of artists returned.
  - c. Please use the Bootstrap Grid system to accomplish the above requirements. Check out [this web page](#) for more detail. Make sure you understand how it

works because it is a great way to realize layouts we learned from the class.

- d. When correctly implemented with Bootstrap, the layout will naturally be responsive (e.g. resizing the browser will automatically resize the images and the nav-tabs, and shift them to the next row).
- e. Typically, iTunes will provide an image URL for each artist. Use the first URL in the list in general.
- f. Note: We provided [screenshots](#) of the search result when a user searches for an artist. When unsure, please refer to these (and the project intro video), your web page **should look pretty much the same** as the one in the screenshot.

### 3. Navigation Tabs (5 points)

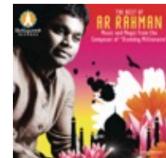
- a. The returned artists should also have their information displayed next to them in the navigation tabs.
- b. Please use Bootstrap's nav-tabs (and other related classes) for this and next part (i.e. 4 and 5). Note that we already provided some example nav-tab code in the starter file. For more info, check out the Hints & Resources section.
- c. The 'Description' tab should contain: artist name, collection name, price, type, and preview link. They can be found in the response JSON data.
  - i. If the price is not provided, set the price to 0.
  - ii. Else, if one of the returned values is an empty string, display "No information provided" instead.
- d. Clicking the preview link should open the preview of the song (or any other media) from this artist in another tab or directly download it (both are fine).

### 4. Track Information (5 points)

- a. When clicking the 'Track Info.' tab, the system should display the Track ID (which is a unique identifier) & Country for that particular search result.
- b. This means, whenever you click on the 'Track Info.' tab, the unique identifier will change for each different 'Track Info.' tab you click on. Here's an example for a query with 'Shreya Ghoshal':



Description      Track Info.  
Track ID: 556951487  
Country: USA



Description      Track Info.  
Track ID: 305196830  
Country: USA



Description      Track Info.  
**Artist Name:** Shreya Ghoshal, Vishal & Shekhar, Vishal Dadlani, Amanat Ali & Raja Hasan  
**Collection Name:** Dostana (Original Motion Picture Soundtrack)  
**Price:** 5.99  
**Kind:** song  
**Preview:** [Link](#)

i.



Description      Track Info.  
**Artist Name:** Shreya Ghoshal, Vishal & Shekhar & Sona Mohapatra  
**Collection Name:** I Hate Luv Storys (Original Motion Picture Soundtrack)  
**Price:** 6.99  
**Kind:** song  
**Preview:** [Link](#)

- c. Hint: The key to getting this section to work is figuring out how to get only that particular div to change when you click on that 'Track Info.' tab. Check out the

Hints & Resources section if you would like a more specific hint.

- d. When clicking back to the ‘Description’ tab, the five types of information should still be there.
5. **Genre Selection (15 points)**
- a. Create a list of tags of which genre labels are available from the search results. (See screenshots/video above)
  - b. As a user toggles a genre, only the artists that are categorized in the selected genre should appear in the grid.
    - i. The number of results should update as well.
    - ii. You can select and deselect genres
    - iii. The genres selected should be indicated by the change in the background color.
  - c. Once you select other genres besides ALL, the ALL button should indicate the change in the background color.
  - d. Please use Bootstrap’s btn (and related classes) for this part. Note that different button colors can and should also be achieved using Bootstrap. For more info, check out the Hints & Resources section.
  - e. If a user has multiple genres selected, the artist shown in the grid should be categorized in one of the selected genres at least (not necessarily all genres selected). (The more genres that are selected the more results should be displayed)
  - f. If ALL is selected, the other genres are deselected and return to the original list sorted by the current sort selection. (see Sort Menu)
  - g. The genre selected should automatically reset to ALL when you search for a new artist.
6. **Sort Menu (15 points)**
- a. A “Sort list of results by: “ button that displays a dropdown menu when clicked.
    - i. There are three options in the menu:
      - Reset to original
      - Collection Name
      - Price
    - b. For “Reset to original”, reset the displayed results to the original order.
      - i. This should be the default selection.
    - c. For “Collection Name”, sort results alphabetically by collection name.
    - d. For “Price”, sort results by increasing price.
    - e. The selected option should be indicated by a distinct background color.
    - f. Please use Bootstrap’s dropdown (and related classes) for this part. Note that different background colors of the options can and should also be achieved using Bootstrap. For more info, check out the Hints & Resources section.
    - g. When multiple genres are selected, the list of results should still be updated by the sort selection (collection name and price).
      - i. In other words, when you select a genre, the sort option should not reset.
    - h. The sort should automatically reset when you search for a new artist.

The following bullet points provide another example of what the sort menu should look like and how it works.

- i. In the dropdown menu, initially, it should be set as the “Reset to original” indicated below by **blue background selection**

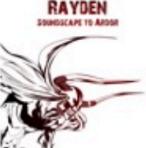
### 493 Artist Search

Enter the name of the artist and enter :

Seach Result (Total 50 Found.)

List of results

Sort list of results by: ▾

Reset to original	Description	Track Info.	Description	Track Info.
Collection Name	 Artist Name: Shiro SAGISU & Yoko Takahashi Collection Name: Neon Genesis Evangelion (Original Series Soundtrack) Price: 9.99 Kind: song Preview: <a href="#">Link</a>	 Artist Name: Rayden & Shiro SAGISU Collection Name: Soundscape to Angor (Remix) - Single Price: 0.99 Kind: song Preview: <a href="#">Link</a>		
Price	 Artist Name: Shiro SAGISU Collection Name: TV Animation BLEACH (Original Soundtrack 1) Price: -1 Kind: song Preview: <a href="#">Link</a>	 Artist Name: Shiro SAGISU Collection Name: Neon Genesis Evangelion (Original Series Soundtrack) Price: 9.99 Kind: song Preview: <a href="#">Link</a>		

- j. When selecting a different option, it should be changed to that selection, for example, selecting “Collection Name”, indicated by the blue background

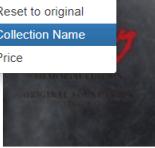
### 493 Artist Search

Enter the name of the artist and enter :

Seach Result (Total 50 Found.)

List of results

Sort list of results by: ▾

Reset to original	Description	Track Info.	Description	Track Info.
Collection Name	 Artist Name: Shiro SAGISU Collection Name: BERSERK The Golden Age Arc MEMORIAL EDITION ORIGINAL SOUNDTRACK Disc 2 Price: 9.99 Kind: song Preview: <a href="#">Link</a>	 Artist Name: Shiro SAGISU Collection Name: BLEACH the Movie: The DiamondDust Rebellion (Original Soundtrack) Price: 16.99 Kind: song Preview: <a href="#">Link</a>		
Price	 Artist Name: Shiro SAGISU Collection Name: BLEACH the Movie: The DiamondDust Rebellion (Original Soundtrack) Price: 16.99 Kind: song Preview: <a href="#">Link</a>	 Artist Name: Shiro SAGISU & Yoko Takahashi Collection Name: Evangelion Finally Price: 10.99 Kind: song Preview: <a href="#">Link</a>		

- k. After an artist search and sorting by “Collection Name”, then selecting different genres should still update the list by “Collection Name”.

In other words, if the user sorts by Collection Name, it should update the list sorted by collection name; after that, if the user picks a genre, e.g. Anime, it should update the Anime list sorted by collection name too.

493 Artist Search

Enter the name of the artist and enter :

Search Result (Total 8 Found.)

[ALL](#) [Soundtrack](#) [Electronic](#) [Anime](#)

List of results

[Sort list of results by: ▾](#)

<div style="display: flex; justify-content: space-between;"> <a href="#">Description</a> <a href="#">Track Info.</a> </div> <p><b>Artist Name:</b> Shiro SAGISU  <b>Collection Name:</b> BERSERK The Golden Age Arc MEMORIAL EDITION ORIGINAL SOUNDTRACK  Disc 2  Price: 9.99  Kind: song  <a href="#">Preview: Link</a></p>	<div style="display: flex; justify-content: space-between;"> <a href="#">Description</a> <a href="#">Track Info.</a> </div> <p><b>Artist Name:</b> Shiro SAGISU  <b>Collection Name:</b> BLEACH the Movie: The DiamondDust Rebellion (Original Soundtrack)  Price: 16.99  Kind: song  <a href="#">Preview: Link</a></p>
<div style="display: flex; justify-content: space-between;"> <a href="#">Description</a> <a href="#">Track Info.</a> </div> <p><b>Artist Name:</b> Shiro SAGISU  <b>Collection Name:</b> BLEACH the Movie: The DiamondDust Rebellion (Original Soundtrack)  Price: 16.99  Kind: song  <a href="#">Preview: Link</a></p>	<div style="display: flex; justify-content: space-between;"> <a href="#">Description</a> <a href="#">Track Info.</a> </div> <p><b>Artist Name:</b> Shiro SAGISU  <b>Collection Name:</b> TV Animation BLEACH (Original Soundtrack 1)  Price: -1  Kind: song  <a href="#">Preview: Link</a></p>

- I. This should work when selecting any of the three sort options.

## 7. BONUS (10 points max)

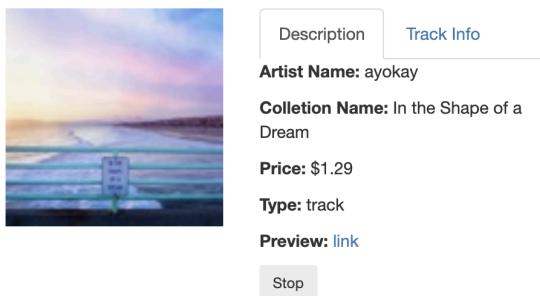
- a. This is completely optional.
- b. If you have added any feature, please include README.txt file in the submission and explain what you implemented; otherwise we won't grade this task (this is the only time you are allowed to add a file to the directory structure).
- c. DO NOT CHANGE ANY EXISTING REQUIREMENTS FROM THE SPEC!
- d. Add a "play" button to each of the 'Description' tabs (as shown below). After pressing the play button, it should start playing the song, and on the button, "play" should change to "stop". When the "stop" button is clicked, it should stop playing the song and revert back to "play".

[Description](#)
[Track Info](#)

**Artist Name:** ayokay  
**Collection Name:** In the Shape of a Dream  
Price: \$1.29  
Type: track  
[Preview: link](#)

[Play](#)

i.



ii.

- e. Add a button after 'Sort list of results by:'. After clicking the button, a popup should appear that displays all the song (track) names returned by fetch grouped by collection name.

## Note about CORS Headers

In doing this project, you might run into your search results being rejected with a “CORS” message. This error is largely server-side, so there is no simple solution. The success of the request will change depending on what browser you are using, if you are hosting the page on the server, and the configuration of the iTunes server the artist you are requesting is stored on.

**One solution that often works is adding "&origin=\*" to the end of your search query. This will often resolve the error if you are running your code directly from a file.**

There are other ways that might help if you still encounter this issues:

- Test with an artist that isn't the one shown in the example video. We suspect that iTunes has changed the way it processes requests coming from the UMich wireless network for the example artist, since there are suddenly several hundred identical requests every minute in the exact same format from the exact same location. Suggesting a different artist to search would be self-defeating, but you should be testing your implementation with a variety of artists anyway.
- Ordering matters in the iTunes API call, so you can play around with the ordering of your added keys to the query
- Do not open the file as a local server (localhost:XXXX/<file path>). My IDE automatically opens HTML files in a server rather than opening the raw file in a browser (<filepath>), and opening it as a raw file stopped it

Again, many of these errors are configuration dependent. If you continue to have issues, please come speak to us in office hours or post on Piazza.

## Hints & Resources

The following tutorial contains a list of programming syntax that *can be* useful for this assignment and the link to their corresponding documentation.

### Vue 3

- Get Started using createApp: <https://vuejs.org/guide/quick-start.html#without-build-tools>
- Vue Cheat Sheet w/examples:  
<https://www.vuemastery.com/pdf/Vue-Essentials-Cheat-Sheet.pdf>
- Template: <https://vuejs.org/guide/essentials/template-syntax.html>
  - Dynamically update HTML text using {{ }}
  - Dynamically update HTML attributes using v-bind
  - Single JavaScript expressions can be used in {{ }} for text and " " for attributes
- Binding Classes (v-bind): <https://vuejs.org/guide/essentials/class-and-style.html>
  - Dynamically toggle a class depending on certain conditions
- Conditional Rendering (v-show, v-if): <https://vuejs.org/guide/essentials/conditional.html>
  - Dynamically render a block (e.g. <div>) under certain conditions
- List Rendering (v-for): <https://vuejs.org/guide/essentials/list.html>
- Events (v-on): <https://v2.vuejs.org/v2/guide/events.html>
  - Listener
  - Method Handler
  - Modifier (prevent page from reloading)
  - Keyboard events (listening for specific keys)
- For making api calls, we can use the following. Both are supported in Vue 3.
  - Axios:
    - <https://vuejs.org/v2/cookbook/using-axios-to-consume-apis.html>
    - <https://axios-http.com/docs/example>
  - fetch: [https://developer.mozilla.org/en-US/docs/Web/API/Fetch\\_API/Using\\_Fetch](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch)

### Bootstrap

- Intro: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- Button (btn): <https://getbootstrap.com/docs/5.3/components/buttons/>
- Dropdown: <https://getbootstrap.com/docs/5.3/components/dropdowns/>
- Navigation tabs: <https://getbootstrap.com/docs/5.3/components/navs-tabs/>
- Grid System: <https://getbootstrap.com/docs/5.3/layout/grid/>

### JavaScript

- Object.keys():  
[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Object/keys](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/keys)
- Object.entries():  
[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Object/entries](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/entries)
- Array.prototype.sort():  
[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Array/sort](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort)

- `Array.prototype.forEach()`:  
[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Array/forEach](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/forEach)
- Arrow functions:  
[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Arrow\\_functions](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Arrow_functions)

### *iTunes API*

- Constructing Search
  - [https://developer.apple.com/library/archive/documentation/AudioVideo/Conceptual/iTuneSearchAPI/Search.html#/apple\\_ref/doc/uid/TP40017632-CH5-SW1](https://developer.apple.com/library/archive/documentation/AudioVideo/Conceptual/iTuneSearchAPI/Search.html#/apple_ref/doc/uid/TP40017632-CH5-SW1)
  - Hint: Most useful parameters are **term** and **attribute**
- Search Examples
  - [https://developer.apple.com/library/archive/documentation/AudioVideo/Conceptual/iTuneSearchAPI/SearchExamples.html#/apple\\_ref/doc/uid/TP40017632-CH6-SW1](https://developer.apple.com/library/archive/documentation/AudioVideo/Conceptual/iTuneSearchAPI/SearchExamples.html#/apple_ref/doc/uid/TP40017632-CH6-SW1)
- Understanding Search Results
  - [https://developer.apple.com/library/archive/documentation/AudioVideo/Conceptual/iTuneSearchAPI/UnderstandingSearchResults.html#/apple\\_ref/doc/uid/TP40017632-CH8-SW1](https://developer.apple.com/library/archive/documentation/AudioVideo/Conceptual/iTuneSearchAPI/UnderstandingSearchResults.html#/apple_ref/doc/uid/TP40017632-CH8-SW1)

### *Additional Tips/Hints*

When trying to toggle the ‘Description’ and ‘Track info.’ tabs, take a look at [this tutorial](#) and play around with the code. Note how this can be achieved simply using Bootstrap and HTML.

Specifically, quoting directly from it: ‘Add `data-toggle="tab"` to each tab, and add a `.tab-pane` class **with a unique ID for every tab** and wrap them in a `.tab-content` class.’

If you have trouble (e.g. clicking on 1 Description tab changes *all* Description tabs, or no matter which Description tab is clicked, only 1 Description tab changes), make sure the stuff bolded is implemented correctly. Put into context of our assignment, each ‘Description’ and ‘Track info.’ tab must have its own unique ID (i.e. `<div id='[something unique]> </div>`). In other words, suppose there are 50 songs returned by the API call, then there will be 50 ‘Description’ and ‘Track info.’ tabs, so we need 50 unique IDs for ‘Description’ and 50 unique IDs for ‘Track info.’

Note: This is not the only way to achieve this behavior, but it is in my opinion the easiest way to do so.