# CPSC 1520 Assignment 2: Top Albums Filtering

## Introduction

This assignment focuses on your knowledge of arrays, loops, objects, and the fetch API.

This is essential because, on most web pages, you fetch some sort of RESTFUL API in JSON and render the result on the page.
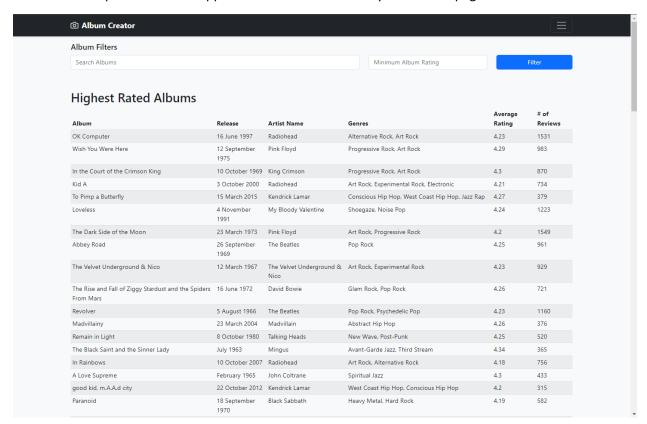
## Overview

For this assignment, you're going to continue the album creator, and you're going to implement a page where users can filter the top albums on the site, which are loaded via a RESTFUL API.

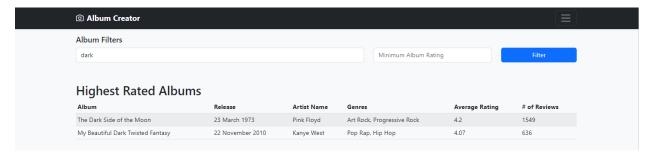Below are screenshots of the application design showing the functionality you'll need to replicate in this assignment.

Please refer to the "**Required Tasks**" and the "**Marking Key**" sections for instructions on completing this assignment.
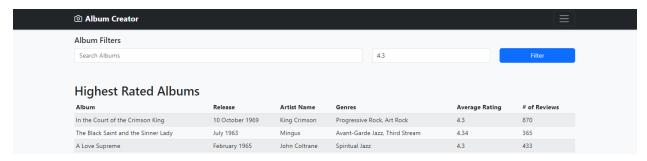
## Sample Functionality

Here's an example of what the application should look when you load the page.

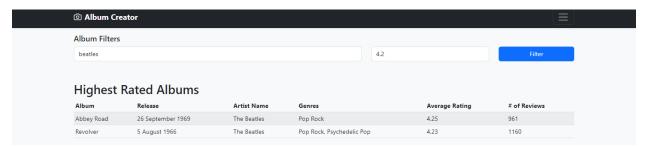| Album | Release | Artist Name | Genres | Average Rating | # of Reviews |
|---|---|---|---|---|---|
| OK Computer | 16 June 1997 | Radiohead | Alternative Rock, Art Rock | 4.23 | 1531 |
| Wish You Were Here | 12 September 1975 | Pink Floyd | Progressive Rock, Art Rock | 4.29 | 983 |
| In the Court of the Crimson King | 10 October 1969 | King Crimson | Progressive Rock, Art Rock | 4.3 | 870 |
| Kid A | 3 October 2000 | Radiohead | Art Rock, Experimental Rock, Electronic | 4.21 | 734 |
| To Pimp a Butterfly | 15 March 2015 | Kendrick Lamar | Conscious Hip Hop, West Coast Hip Hop, Jazz Rap | 4.27 | 379 |
| Loveless | 4 November 1991 | My Bloody Valentine | Shoegaze, Noise Pop | 4.24 | 1223 |
| The Dark Side of the Moon | 23 March 1973 | Pink Floyd | Art Rock, Progressive Rock | 4.2 | 1549 |
| Abbey Road | 26 September 1969 | The Beatles | Pop Rock | 4.25 | 961 |
| The Velvet Underground & Nico | 12 March 1967 | The Velvet Underground & Nico | Art Rock, Experimental Rock | 4.23 | 929 |
| The Rise and Fall of Ziggy Stardust and the Spiders From Mars | 16 June 1972 | David Bowie | Glam Rock, Pop Rock | 4.26 | 721 |
| Revolver | 5 August 1966 | The Beatles | Pop Rock, Psychedelic Pop | 4.23 | 1160 |
| Madvillainy | 23 March 2004 | Madvillain | Abstract Hip Hop | 4.26 | 376 |
| Remain in Light | 8 October 1980 | Talking Heads | New Wave, Post-Punk | 4.25 | 520 |
| The Black Saint and the Sinner Lady | July 1963 | Mingus | Avant-Garde Jazz, Third Stream | 4.34 | 365 |
| In Rainbows | 10 October 2007 | Radiohead | Art Rock, Alternative Rock | 4.18 | 756 |
| A Love Supreme | February 1965 | John Coltrane | Spiritual Jazz | 4.3 | 433 |
| good kid, m.A.A.d city | 22 October 2012 | Kendrick Lamar | West Coast Hip Hop, Conscious Hip Hop | 4.2 | 315 |
| Paranoid | 18 September 1970 | Black Sabbath | Heavy Metal, Hard Rock | 4.19 | 582 |

Below is what the application should look like if you search for "dark".
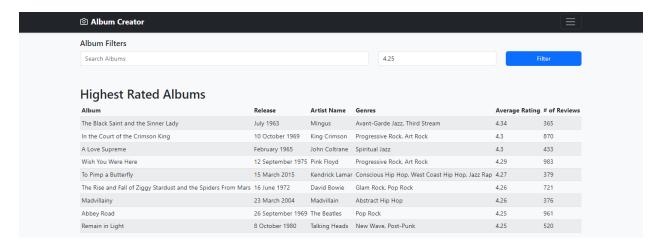


Below is what the application should look like if you have a minimum rating of 4.3.



Below is what it should look like if you search for "beatles" with a minimum rating of 4.2.



Bonus part if you click on the "average rating" or "# of reviews" column headers it should sort by that date.

## Required Tasks

- Part 1: Fetch the albums and render them on the page.
    - Create a function to get the album data from our local server.
        - Use fetch API (using either the promise syntax or the async await syntax) in a function and return the fetched data from that function.
        - Reassign the variables "topAlbums" and "topAlbumsFiltered" to the data that was fetched from our REST API. This can be done in the function defined above or in a separate function that should be called when the page is loaded.
        - The data should be loaded when the page is loaded or refreshed.
    - Render the Albums on the page.
        - Use map to loop through the arrays of objects that are fetched.
        - Append the table rows to the correct part of the HTML table element.
- Part 2: Albums can be filtered by the search query and the minimum rating.
    - Add an event listener and handler to the form.
    - Get the search query and minimum rating values from the inputs the same we've been doing throughout the course.
    - Create a couple of functions that will filter the minimum rating and the album names.
        - The first function should filter the albums on either the "Album Name" or "Artist Name" properties on the album data that has been fetched. The search functionality should be case-insensitive.
        - The second function should filter the albums on the "minimum rating" property on the talks that have been fetched. The talks shown should have a greater rating than what is entered.
    - Using the functions created in the last point, filter the album data and render them on the page. Quite a few examples above show you what this should look like.
- Bonus Part: Sorting Average Rating and Minimum Reviews
    - If you click on the Average Rating and Minimum Reviews heading, it should sort from highest to lowest.
- Challenge Part: (not for marks): Sort by Release Date
    - If you click the Release Dates heading, it should sort from highest to lowest
- The code style will be specified by your instructor.

## Marking key

| Tasks | Grade | Marks | Total |
|---|---|---|---|
| **Fetch and rendering.**<br>• Fetched Album data successfully, correct syntax.<br>• Album data rendered properly.<br>• Map used for looping over the talks.<br>• topAlbums and topAlbumsFiltered variables are reassigned properly as the result of the fetch call back. | | 5<br>3<br>1<br>1 | |

| | | | |
|---|---|---|---|
| **Loops and Filtering**<br>    • Successfully added event listener and prevented the form from submitting.<br>    • Search query filtering function (not case sensitive) filters album data correctly.<br>    • Minimum Rating filtering function filters album data that has more than the value specified by the user.<br>    • The filtered album data is rendered correctly.<br>    • Filtering functions will only be called if the inputs have values. | | 1<br>5<br>5<br><br>3<br><br>3 | |
| **Bonus**<br>    • Sort columns ascending if the columns are clicked | | 3 | |
| **Code**<br>    • Code Formatting and Style<br>    • Errors running the project | | -3<br>-3 | |

# Marking Rubric

| Marks | 5 Marks Criteria |
| --- | --- |
| 5 | Task was completed with the highest of proficiency adhering to best practices and followed subject matter guidelines all tasks were completed to a professional standard. |
| 4 | Task was completed well some minor mistakes. Well above average work shows good understanding of the task and high degree of competence |
| 3 | Satisfactory work some features missing or incorrectly implemented. Show a moderate level of understanding in the task with room for improvement. |
| 2 | Below average work. Task was poorly complete. Show understanding of the task and the requirements to implement but implementation was poorly executed. |
| 1 | Some of the task was completed. Showed a lack of understanding in the subject matter and very poorly executed |
| 0 | Not completed. |

| Marks | 3 Marks Criteria |
| --- | --- |
| 3 | Proficient shows a high degree of competence in completing task. |
| 2 | Capable above average degree of competence in completing task |
| 1 | Satisfactory shows a satisfactory degree of competence in completing task. |
| 0 | Shows a limited degree of competence in completing task. |

| Marks | 1 Marks Criteria |
| --- | --- |
| 1 | Task Completed satisfactorily |
| 0 | Task was not executed. |