***VaynerMedia Data Science Front-End Challenge!***

We are going to build a small UI for sorting ALBUMS for two different USERS!

Stuff you’ll need

jQuery =><http://jquery.com/download/>

Mock-API docs =><https://jsonplaceholder.typicode.com/>

Don’t use any packages or add-ons besides jQuery for any part of this challenge. You are welcome to write in vanilla JavaScript without jQuery, but it isn’t recommended. Anything supported by Chrome is acceptable JavaScript.

Your submission should be a single zip file, and include an html, js, and css file (and also a file for the jQuery package, if you use it). I should be able to unzip the file, then drag the html into a Chrome window, and see your work in action.

**Step 1 - Request**

GET USERS with id 1 and 2 from the mock-API. Also GET ALBUMS associated with these two USERS.

**Step 2 - Display**

Represent both USERS in two tables. Both tables should have the name of the USER above it, and each row in both tables should represent an ALBUM that belongs to each USER, and each column should represent a property of the ALBUM (Id and Title). **No <table> tags please!**

**Step 3 - UI**

Implement drag-and-drop. You should be able to drag a row from one table, and drop it on the other table. Upon doing so, the UI should display the row in the dropped-on table, and no longer in the dragged-from table.

The drop action should perform an AJAX request to update the ‘userId’ property on the ALBUM. The request will not actually persist in the api, so we are just going to pretend it was successful. When the request resolves, you should use the **response** data to reflect the change in the UI. Do not fake the response data, wait to receive it, then use it to update the UI.

**Step 4 - Styling**

Use ‘display: flex’ instead of floats/margins for alignment and styling at least 90% of the time. Margins may be used in concert with flex if you’d like.

Each table should be large enough to accommodate its initial contents. When items are dragged from table to table, overflow should be scrollable.

Rows in the table should be candy-striped (every even row should be white, and every odd row a color of your choice). Be creative!