## This challenge is solely for the purpose of interviewing and any candidate submissions will not be used by BoviSync in any way.

## Create a mock animal page

There is a part of our application where we display user-defined data about a specific cow, chosen from a list of a few thousand potential queries, which we call “report items.” This exercise has you create a customizable view using some data similar to what you would see in our production system. We expect this project to take you between 1 and 4 hours. If you think that this project will take you longer than that time, please provide a detailed estimate of the time it would take to create this sample application, and use your judgement on completing the parts you think are most critical in roughly 4 hours.

You are welcome to use any open source JavaScript or CSS frameworks or libraries you like. Deliver this sample to us using either a hosted git repository on GitHub, Bitbucket or GitLab or online code editing tool (like CodePen or Plunker). If delivered as a git repository be sure to include a readme with instructions on how to run the application.

You will use data from 3 JSON files for input to your sample. This data should be fetched using the JavaScript “fetch” API or a HTTP library of your choice. You may fetch them directly from the URLS below or place them with your code.

## Input Data

### Report Item meta - <https://bovisync.bitbucket.io/sample_data/item_meta.json>

This is a list of available report items along with some basic information about them, including a description and data type. The possible data types for the report items are “string”, “integer” and “decimal”, as seen in the JSON file.

### Page meta - <https://bovisync.bitbucket.io/sample_data/page_meta.json>

This is the user customizable list of sections to display on the page, which defines which report items are to be displayed

### Animal Data - <https://bovisync.bitbucket.io/sample_data/animal_data.json>

This is the specific data for each requested report item.

## Page Requirements

Display the data from animal\_data.json as described in page\_meta.json, using item\_meta.json as needed for labels or other metadata. The item “shortName” is universally unique and can be considered the primary key for the items. The application should be flexible enough to support arbitrary items defined in the same format as the provided item meta (within the 3 data types provided).

The page meta may contain up to 10 sections of data, and up to 10 report items within each section. Each section should be visibly distinct on the page and include the label defined in the JSON. The UI is your decision, some possibilities are tabs, cards, accordion, grid sections, etc.

Each report item should display the item name and provide a means for the user to see the description and short name. The description and short name should not be shown without some kind of user input (click, hover, etc). Some ideas for description display are a tooltip, modal or initially hidden element.

Numeric items should support conditional formatting based on a single user-defined threshold defined per item. Item values above the threshold value should stand out to the user in some way. The threshold can be set through a UI element or defined in a new JSON file or documented code reference.

Decimal display values should round to a user-defined number of decimal places per item, without losing the underlying precision (so for example we could perform a calculation with the value at full precision). Like the threshold, the number of places to display can be defined either in the UI or code.

Gracefully handle items requested in the page meta which are not present in the item meta or animal data. Provide the user with useful feedback and avoid interfering with display of other data.