

JavaScript Coding Exercise

Overview

Brightwheel uses internal and third-party APIs extensively throughout our platform to process and store data, as well as manage communication to end users. The goal of the exercise will be to consume a live API and display and update the data according to the specifications given below. This exercise is meant to assess your overall frontend skillset and your design aesthetic.

The Exercise

Build a **single-page web app** that uses the provided API to create a search engine landing page with the ability to “star” results.

Provide us the link to your final product as a cloneable repo on Github or Bitbucket

Implementation Requirements

This will be a simple exercise, but structure and organize your code as if it were going into production. It should be fully-functioning code that can be run.

We have provided a local backend to use, and an empty React app that you may use as a starting point if you wish. Documentation on the API and how to run it can be found in the project README.

The API provides mock data about Companies, Animals, and Products. Each of these types should be rendered to show the relevant details as outlined below.

- There should be an Input field you can type in
- The results should be displayed as you type
- Show the first 10 results below the search input
- Format the search results as follows:
 - For `Products`, show name, product category, preview text and image (if available)
 - For `Animals` show animal name, scientific name and image(if available)
 - For `Companies` show name, description and address
- Clicking on a result should toggle the starred/unstarred state for the result and should make an API call to update the record in the database.
- There should be a counter that displays the total number of starred results in the database

We'd prefer to see you use React-based features like context and state over Redux, but use any component or style libraries you like to showcase your ability to write clean, reusable code. We

want to see your approach to handling UI-state and business logic as well as your design choices and ability to write semantic and accessible HTML.

This exercise should take about 2 hours. If you have extra time, bonus points for:

- Showing a complete list of starred results when user clicks on the star count
- Component or integration tests

How we will Review

- **Functionality:** Does the app do what we asked?
- **Code Quality:** Does the code follow software engineering standards and best practices? Is it easy to understand and maintainable?
- **Technical Choices:** Is the implemented technical approach appropriate for what was asked?
- **Design Choices:** How does it look and feel?