Photos Grid App

Task: Web Application (Angular/React.)

The importance of AJAX and JSON is growing exponentially on the web. This exercise is aimed at testing your knowledge of Angular (or ReactJS), JavaScript, UI development, and your creativity.

For this example, we will be using a simple image API.

For this exercise:

1. Make an AJAX GET request to https://raw.githubusercontent.com/BuyProperly/interview/main/photos/animals.json **(This contains some Extra points for using the Angular HTTP client with Observables).**
2. Format the results into a list. Each item must include a **title**, **description**, and a small **thumbnail** version of the image. The list can be displayed vertically, horizontally, or in a grid, however, you wish.
3. Create a **modal** dialog that displays the full-sized version of an image and any additional details provided in the JSON (e.g. Family, Genus, Collective Noun). The modal should appear after clicking on a thumbnail.
4. Create two navigation items (or buttons) at the top of the page - “Animals” and “Fruit & Veg”.
5. Repeat steps 1-3 for the following URL: https://raw.githubusercontent.com/BuyProperly/interview/main/photos/fruitveg.json
6. Make the two navigation items in the top bar load the appropriate lists.
7. Create a user interface using CSS. When creating your user interface, use the BuyProperly website style as a guide (https://buyproperly.ca).
8. Optimize your code as much as possible for performance and readability, adding comments where necessary.
9. Structure and set up your working directory in a manageable fashion, and organize your folder hierarchy.

Notes:

1. If this task will be built with **Angular**, we *strongly* recommend using **Angular CLI** (https://angular.io/cli) to set up your project. This sets up the barebones for an angular project, including development and production build scripts, making it simple for you to work on locally, then host on GitHub or send to us.
2. If this task will be built with **React.js**, we *strongly* recommend using **create-react-app** (https://reactjs.org/docs/create-a-new-react-app.html) to set up your project.

Bonus points (optional)

1. Make use of the newer language features of JavaScript (ES6/ES7) in your code.
2. Use a CSS preprocessor(Sass/Less) library to write your styles.
3. Using **ngx-store/recoiljs** to store the JSON values in persisted reactive state.

Note: If you experience CORS errors while using Google Chrome, we recommend testing your application with Firefox or a recent version of Internet Explorer.