Encord Frontend Technical Challenge

Introduction

We ask that you spend no more than 3 hours on this task, if time becomes an issue, please focus on finishing existing functionality.

What we're looking for:

- Functionality matching the brief
- Consistent code style
- Good code structure, patterns & abstraction
- An intuitive and pleasant to use UI

Using either React or Next.js and Typescript, please build a React-based Single Page Application to satisfy the brief specified below. Please feel free to bring in any additional frameworks or dependencies to help in development. We may just ask you to explain these choices. Please also feel free to leave comments in the code in areas you may want to highlight if you'd want to tackle it differently if time weren't such a consideration.

We ask you to use https://www.npmjs.com/package/json-server to stub out the API request this application will make. The JSON and image to be used for this application are provided at the end of this document. No other APIs should be integrated into this app.

The Brief

The application has 2 tabs: Images & Predictions:

Images

This tab gives the user capability to upload into memory (no need to call an API here) and manage a list of images which will be used for further processing. The list of images should be displayed in a tabular format with the following columns:

- Filename of the Image
- Size of the image
- Time of Upload
- A button called PREDICT

On clicking the PREDICT button:

- A dialog box asking for 'Title' and 'Description' of the prediction should be displayed - Dialog box should have option to Submit and Cancel
- On clicking submit the request to the JSON server should be made, saving the response against the image.
- On successful completion of the API request a single new entry would be added to the Predictions tab.
- On a failing API request, show an error message to the user

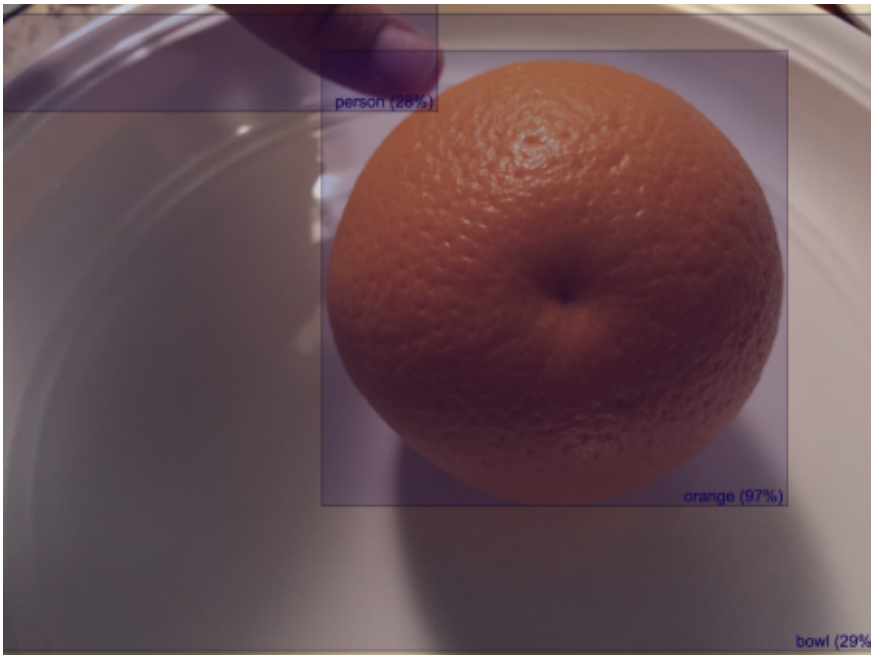
Predictions

This tab gives users the ability to view the predictions generated by this app. The list of predictions should be displayed in a tabular format with the following columns:

- Title (entered by the user in the dialog)
- Description (entered by the user in the dialog)
- Formatted timestamp of running the prediction
- Button called VIEW

On clicking the VIEW button

- The image on which the prediction was run should be displayed.
- All the items in the prediction of this image should be displayed.
- A transparent rectangle covering the coordinates in the prediction item should be displayed over the image for all items predicted within an image, the image and prediction should be responsive.
- The “label” and “score” should be displayed along with the prediction for the item. Please refer to the image below for inspiration.



The above image shows 3 Prediction items using transparent rectangles having coordinates returned by the API

We wish you the best of luck and look forward to your submission. Thank you for your time!

Assets

Image of Orange:
https://drive.google.com/file/d/11MvR5va466svjUxQIrtzhDhgTQ3FIeUJ/view

db.json for JSON-server. This has a hard coded response for the `/predict` endpoint.
https://drive.google.com/file/d/1sBb04PN-VzzWwZksrsMeSQFKuaQgmcGz/view?usp=sharing